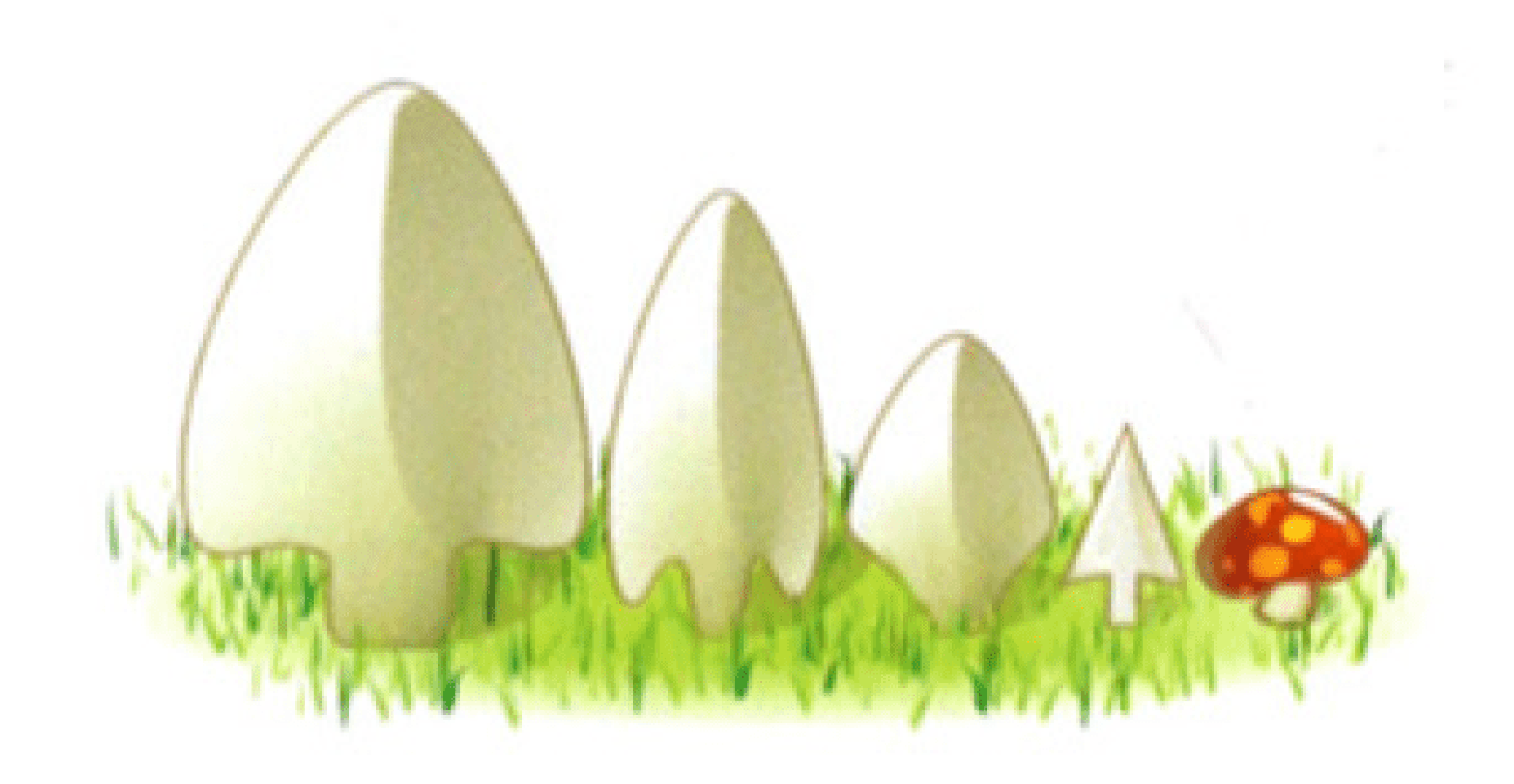
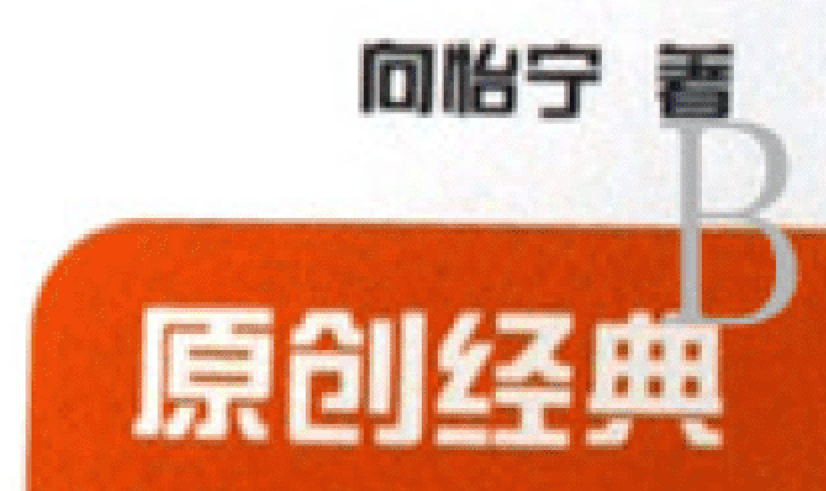
就这么简单

——Web开发中的可用性和用户体验





向怡宁 著



清华大学出版社

就这么简单

——Web 开发中的可用性和用户体验

向怡宁　著

清 华 大 学 出 版 社
北　京

内 容 简 介

本书主要讲述人机交互学（Human-Computer Interaction，简称 HCI）这一学科在 Web 界面设计领域中的应用。

正如你所看到的，这是一本文字轻松，而且有很多好玩插图的……技术理论书籍。

没错，技术理论。它将和你讨论关于软件（或网站）的界面与用户之间的关系——界面是你做的，而使用它的是用户——所以，别自信满满地说自己做得很牛，让我们听听用户的意见。这本书能告诉你这一套顺序大概是怎样的，你应该注意些什么，最终水到渠成的是一个能经得起考验的产品。

这本书可以说是一本针对广泛大众，以及那些没有接受过人机交互和可用性专业培训的用户界面设计人员的辅导和帮助手册。你不必背诵那些拗口的专业词汇，你只需知道“我该如何去做”从而解决问题。这本身也是人机交互学科的一个重要原则——“易于学习，从而便于使用”。

你明白了。这不是本情节跌宕起伏的小说，也不是行文优美如诗的散文。但我仍然希望它能给人带来快乐——学习不是一件枯燥的事，但很多人把它变得枯燥了。绝大多数技术理论书籍的作者们都正襟危坐就好像板着面孔的老师，不可否认，对待学术问题应当态度严谨，但是严谨不代表枯燥，能在专业知识课堂上把大家逗笑的老师一定不简单。

我不敢说我做到了，但我在努力。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

就这么简单：Web开发中的可用性和用户体验/向怡宁著.—北京：清华大学出版社，2008.6

ISBN 978-7-302-16999-4

Ⅰ.就… Ⅱ.向… Ⅲ.用户界面－设计 Ⅳ.TP393.092

中国版本图书馆CIP数据核字（2008）第017158号

责任编辑：陈 冰
责任校对：徐俊伟
责任印制：
出版发行：清华大学出版社　　地　址：北京清华大学学研大厦A座
http://www.tup.com.cn　　邮　编：100084
社 总 机：010-62770175　　邮　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　销：全国新华书店
开　本：203×260　印　张：20　字　数：530千字
版　次：2008年6月第1版　印　次：2008年6月第1次印刷
印　数：1　000
定　价：　元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　产品编号：027405-01

业界专家对本书的评语

本书从最日常的界面使用体验出发，简单易懂，尤其是结合了实际工作案例进行讲解，是一本很难得的国内交互设计的入门读物。

——Rokey

Eico Design 创意总监，国内最知名 GUI 设计师

这大概是国内第一本非“学究”性的书籍，全面介绍了 **UCD** 的相关知识。其内容易于理解，对于向更多的人传播 **UCD** 思想很有意义。期待看到更多类似书籍在国内出版。

—— 白鸦

产品设计顾问，可用性专家

纵览本书，无论你是处于产品开发流程中哪一环节的人员，你都会发现它对你有所帮助。作者描述了 **UCD** 的各种方法，通过有趣的介绍形式，把你引领到 **UCD** 的领域中来。

——Peter Qu

联众公司 UE 工程师

敞开心扉，转换思路，跟着这本书去领会怎样去设计才能真正服务于用户。以用户为中心的概念，的确就是这么简单。

—— 田明慧

SIEMENS 中国研究院　用户界面设计中心经理

用通俗易懂的语言，结合实际的案例，由浅入深的介绍了 **UI** 设计，帮你从交互设计的角度重新认识 **Web** 设计。

——H ++（火山大陆）

微软亚洲设计中心 User Experience Designer

编辑的话

这是我来清华后策划的所有书中最有趣的一本，事实上，准确地说，它是目前所有已出版的计算机书中极为有趣的一本。而我，很荣幸是它的策划编辑。

在介绍这本书前，我觉得有必要说说这本书的作者向怡宁，这是一个瘦瘦高高的扎着辫子的有趣的人。很酷。是个人才，事实上，准确的说，是个全才，因为他在美术（GUI 设计师和漫画家）、音乐（布鲁斯口琴和摇滚乐队的吉他手）、文学（魔幻小说和计算机书作者）等诸多方面都有所造诣。当然，有时，这多多少少会让人有些恼火。

说到这本书的诞生，至少从我的角度上看，是很顺利的，过程大致如下——

有一天，他对我说，“陈冰，我想写一本关于 Web 用户体验的书”，我说，“好啊”，于是，他就回家写去了，写呀写，写呀写，几个月后，他把写好的书拿来给我看，“哇，好书”，这就是我当时的评价。一本好书就这样诞生了，看似平淡的几乎令人感到乏味，但这只是站在我的角度看的。

当我站在作者的角度看时，另外一番景象出现在我的脑海中——夜以继日的挑灯夜战，没有白天和黑夜，只有思想的烈火烧灼键盘的声音。

这本书也是我没有提出任何修改意见的一本书，因为实在看不出有什么需要修改的，一切都很完美。

说到这本书的特点，首先，非常幽默，绝对会带给你快乐，这年头能带给人快乐的书太少了，而且更为重要的是，这本书还非常有用，说它有用，主要是因为它里面饱含着思想，大把大把的思想，绝对都是货真价实的东西，就像那种大皮沙发一样，里面塞满了实实在在的材料，都是可以用到你的 Web 开发中的。

还有非常值得一提的一点是，这本书中所有的插图都是作者自己创作和手绘的。全都是漫画的形式，每一幅都很有趣，我经常看着就乐。好玩。不过挺让人恼火。

因为这本书实在太有趣了，以至于我终于忍不住从百忙中抽出时间来亲自做了这本书的文稿编辑，主要是想一睹为快，哈，而且把其中那些我认为很经典的句子都勾画了出来，并嘱咐排版人员一定要用另一种颜色来明显的标示出这些被我勾画出的经典句子，因为我们大家看书时都有一个习惯，就是喜欢把好的和经典的句子都用笔勾画出来，所以现在我干脆直接采用颜色来标示出这些经典句子，就省得读者们以后再自己费劲儿画波浪线了。这样以后翻阅时也很容易看到它们。

你可能注意到了，这本书没有前言，不要奇怪，因为这本书的“序章”，事实上，就是这本书真正的“前言”。

“好”不能轻易得到，“好”就意味着大量的工作。为了给这本书确定一个好的版式，作为这本书的策划编辑，我对书的开本、目录的设计，以及正文字体的选择等等都花费了大量心思，几

经改变和推翻。仅以确定正文使用何种字体举例，最初，为了配合这本书轻松幽默的风格，我选择了幼圆这种视觉上轻松活泼的字体，但后来我发现幼圆字体虽然与书的风格更匹配，但却会导致阅读速度的降低。因为人眼对圆体字的识别速度比对宋体的识别速度要慢，在长时间大量阅读时，这种较慢的阅读速度会让人产生一种压抑感，而这是阅读时最要不得的感受。因此，我重新把正文字体由幼圆替换为宋体，虽然只是一个小小的变动，但却足以影响整本书的阅读体验。

好了，关于这本书，我该介绍的差不多就是这些了。如果你在阅读本书时有任何意见，可以直接联系作者(www.messbook.net)，也可以到我的博客(w3cbook.blog.sohu.com)上给我留言。

最后，我想说，对于任何有价值的事物，不能轻易的兴起，一旦兴起，更不能轻易的废弃。有一天，当一个真正的作者决定写一本好书时，这本好书，实际上，就已经诞生了。

陈冰

本书策划编辑、高级程序员、科普作者

《Flash 第一步》、《电脑使用说明书》作者

目　录

序章　关于这本书

第一章　你首先要知道的一些事情

第三章 设计方案和制作原型

第四章　以用户为中心的开发

第五章 文化的差异

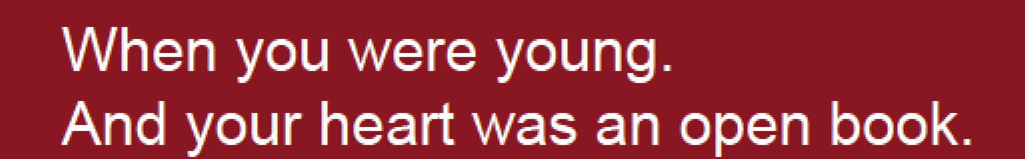
第六章　网页的可用性

第七章 Web-based 产品

第八章 测试可用性

序　章

关于这本书



When you were young.
And your heart was an open book.

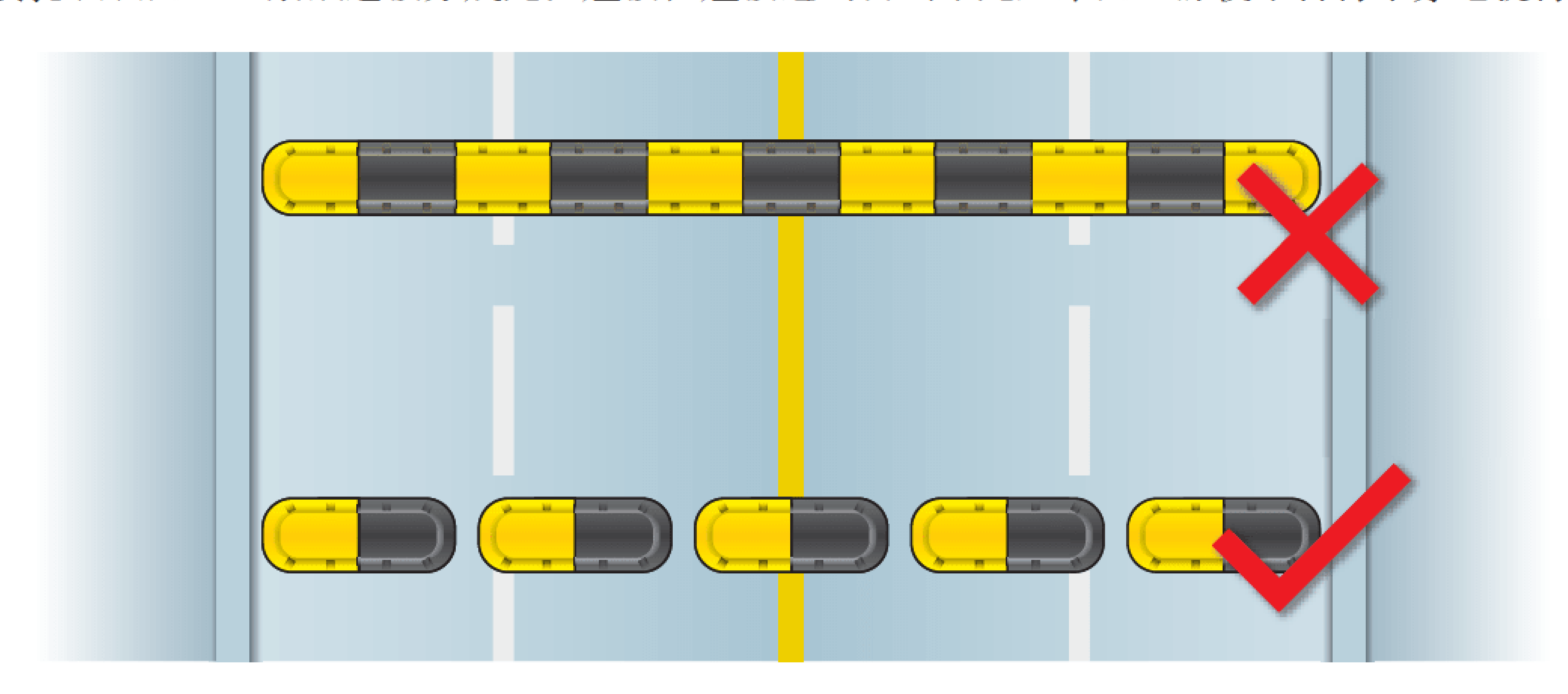
What does it matter to ya?
When you got a job to do,you gotta doit well.

——Paul McCartney

没错，你看到的是一本关于人机交互的书。

如果你还不太了解这本书要讲什么东西……不知道你在住宅区或学校里面看到过减速板没有。它是横在道路中间凸起的一道铁板，用于限制机动车的速度。这是个增强了路人安全性的很好的办法。但是，它在限制汽车的同时却又难为了自行车。在很多小区或者校园，我发现几乎所有的减速板统统是一长条铁板，那些骑自行车的人就只能溜边，从减速板和马路牙子间的狭小缝隙中穿过。

这种糟糕的设施就是没有充分考虑“用户”的结果。其实要限制机动车而不阻碍自行车，只需要把长长的一整条减速板分成几块短板，短板之间留出十几厘米的空隙便于自行车穿过就行了。



这就是是否“人性”的思维方式。在人机交互设计中，也就是我们所说的“可用性”和“用户体验”。网页也好，软件也好，或者是手机、影碟机等设备，只有充分考虑用户在使用时的感受，才能开发出令人满意的产品。

如果你不考虑用户，或者说没有考虑所有的用户，那么就有可能会出现类似下面这种情况 ：



在你打开的这本书里面，我们主要讨论的是网页产品的界面设计。这包括 ：

- **Web，也就是网站。**不管是个人网站还是商业网站，你都需要考虑用户体验和可用性问题。
- **Web-based 产品。**所谓的 Web-based 也就是基于网页的产品或服务。它们不用安装客户端，用户以网页形式就能登录使用。例如网络管理系统或一些在线软件。

简单来说，实现更好的“可用性”和“用户体验”就是我们要谈论的内容——如何才能充分考虑用户的感受？

0.1　谁该看这本书？

先看看你自己。如果你是 ：

- **一个界面美术设计师**，使用 Photoshop 和 Flash 软件来展现自己的创意和美学素养，考虑材质、渐变和光影效果以及页面的布局，但却不太清楚最合用户心意的到底是简洁实用还是美观大方 ；
- **一个软件程序员**，对 Java、C++ 和 ASP.NET 都很熟悉，戴着眼镜写出一行行漂亮的代码。你觉得对软件来说功能的强大和完备才是最重要的，从没怎么在意过用户怎么使用它 ；
- **一个页面制作员**，使用 Dreamweaver 把 100 个表格嵌套在一起，随心所欲地通过 CSS 样式表来控制页面的视觉效果。你认为自己掌握了不少知识和经验，如果有机会可以往更高的层次发展 ；
- **一个产品经理**，甚至是一个网站或网络软件公司的老总，你知道自己手下的技术都很厉害，但不知为什么最后的成品总是有点无法让你安心。或者你想让自己推出的产品或服务能够有更大的竞争力，获得更多的好评从而占有更大的市场份额 ；
- **一个初出茅庐的本科生或研究生**，你掌握了很多人机交互的专业知识，了解很多这方面的技术，但不知道如何才能把你的本领发挥出来，并且告诉上司你做的工作并不比程序员们简单或者低级 ；
- ……

如果你是上述这几类人的其中之一（可能有些家伙还身兼多职），而且 ：

你想了解的是人机交互可用性方面的原理和根本，希望能够真正地把产品变得更好，而不是只记一些技术用于夸耀。

你喜欢的是轻松愉悦的氛围，愿意在睡觉前或者餐桌上看看相关书籍（甚至作为一种消遣），但却很反感枯燥乏味的专题技术报告。

那么我很高兴这本书非常适合你。如果你还在书店里犹豫……赶紧买下来吧！

但是，如果你满足以下任何一种情况：

你没有接触过界面设计，比如没有开发过软件、没有设计过网站，或者不是一个网络产品项目的管理者，你甚至不知道设计界面需要什么软件。

你是一个很有经验的人机交互设计师，希望看到各个交互设计流派的讨论和争吵、最尖端的用户测试技术（例如眼动跟踪系统），此外你还希望得到大量的最新科研成果实例。

你不喜欢跳出框框的东西，觉得技术就是技术，说一不二，只有严谨的词汇才能准确地表达对科学的认识。你觉得一本讲人机交互的书居然是全彩印刷而且满是漫画，这一定不靠谱。

那么，很遗憾，这本书不适合你，再看看架上其他的书吧！



0.2　这是本什么样的书？

我们仍然需要知道，人机交互是一门严谨的学科。它涉及到很多方面的知识，包括人体工程学、软件工程学、心理学、美学、计算机科学、社会科学……很多很多。这个领域在西方研究得比我们要早，大概在上世纪70年代就已经发展成为一门正式的学科了，到今天也已形成了无数的理论和流派。市面上你能找到很多交互设计方面的书，它们绝大部分都非常有用，讲述了很多人机交互的内容，而且非常专业。

是的，非常专业。有时候甚至专业得没有人愿意看下去。不知道你有没有这种体验，就是面对着那些艰涩难懂、通篇专业术语的长篇累牍，明明都看在眼里，却没有经过大脑，然后在十分钟之后把它们全部忘光。过于专业的词汇会让你的大脑有一种潜在的抵触情绪，它会欺骗你的眼睛，让你以为已经“看过”了，但其实你只是“看到”了，而没有形成印象。



……我只是开个玩笑

在这本书里，我不想让你感觉责任重大。毕竟我们的目的只是为了让我们的产品有更好的表现，能够让用户们更方便地操作，从而伸出他们的大拇指：“这些家伙干得不错”。

所以我不想和你谈论过多技术方面的问题。不论你是一个界面美术设计师，还是一个程序设计师，或者是产品经理，甚至是公司的老总，其实你所需要关注的并不是实现的手段如何，或者到底使用了什么技术，你所应该关心的只是“目的达到了没有”。

这是因为，用户不会去在乎你是用HTA技术还是GOMS技术来分析他们的典型任务，或者摆在他们面前的是PICTIVE模型还是CARD模型。他们根本不会考虑这些问题。他们只需要感觉你的产品好用、方便、舒服，他们就会高兴，那么你的目的就达到了。

因此，我希望在你面前的这本书能够告诉你一些需要知道的东西，而不是要逼你了解所有的交互设计技术。我们需要了解必备的常识，然后要保证自己不会忘记。填鸭式的硬塞可达不到这

种效果——除非你是一个自虐式的学习狂人。我希望你能够轻松地学到知识，而不是强迫自己的大脑去死记硬背。



你怎么看待知识？

要想做到这一点，我总结了一些方法 ：

- **多给你一点好玩的图片**

有人说“对大脑而言，一幅图顶得上 1024 个字。”与单纯的文字相比，图片更能让人看得懂、记得住，学习效率也会更高。你的大脑会喜欢一些新鲜的玩意，也许它们只是让你感到有趣、奇怪或者幽默，但你的大脑会投入更大的注意力，它会觉得这些东西更有可能要被记住。也许画图会耗费我更多的时间和精力，但为了让你高兴和轻松，我觉得这是值得的。此外我对自己的联想和绘图能力感到很骄傲，也很愿意把这种能力展示出来。

- **用一种交谈的方式**

我可不想用干巴巴的语调来介绍知识。照本宣科地作报告应当更容易，但那样很没意思，也不会提高你的积极性。想想看，与一个朋友交谈和坐在一堆人中间听报告，哪一种会让你更想打瞌睡？其实你的大脑并不在乎“交谈”的对象究竟是谁，即使是一本书，这对它来说也没什么太大分别。所以我会用很通俗的语言来描述哪怕是很艰深的知识，而且不会板着面孔，偶尔可能会开些玩笑。

- **引起你的注意，并保持你的注意**

你肯定有过这样的体验 ：“我真的很想把这个学会，但是看过一页之后我实在是昏昏欲睡”。你的大脑注意的是那些不一般、有意思、有些奇怪、意料之外的东西。学习一项有难度的新技术并不一定枯燥。所以我可能会使用一些方法来引起你大脑的“警觉”，让它做一些评价和判断，

而不会老是呆在那里打瞌睡。

- **有所取舍，才能有所收获**

这本书的目的是让你了解交互设计“可用性”方面的知识，从而使你的工作（不管是结构设计、程序开发还是美术设计）更系统，更容易达到好的效果。之前我已经说过，介绍技术不是这本书的重点。如果你真的要成为人机交互设计的大师，这本书肯定也不会是你唯一关于交互设计的书。所以我不打算方方面面都照顾到，以免弄成了“贪多嚼不烂”。这里只提供你真正需要的东西。

- **我不布置作业**

在上学的时候作业已经做得够多的了。这本书不会让你感到自己必须付出多少劳力来应付老师（如果存在一个“老师”的话）。很多书提供了各种各样的习题，但没有提供答案。这种习题除了让我们对自己更加怀疑之外没有什么作用。还有很多书在习题下面就是答案。既然答案就在下面，为什么不把它们以更好、更容易接受的方式呈现出来？真正的习题是我们在现实中遇到的具体案例。

在这本书里，我尽力做到系统、流程化地描述交互设计的方方面面。如果你在某个环节遇到了问题，翻翻对应的章节。翻个两三次你就记住了。我们不需要习题。

0.3　怎么用这本书?

前面已经说了这么多，下面就看你的了。我在这里提出了一些建议，你可以看看，然后听听你的大脑怎么说，弄清楚对你来说哪些做法可行，哪些做法不能奏效。当然你也可以做一些新的尝试。

- **看慢一点**

毕竟你不是在看小说。也许你会觉得有些意思，但如果只是随便翻翻，不用心去理解的话，可能也没有多少效果。慢一点，有些东西需要想一想。如果在书里感觉有些疑惑，先弄清楚再接着读，而不要把它放在一边“待会再去考虑”。我希望你能读进去，越深越好。要知道，**理解的越多，需要记住的就越少**。

- **大声说出来**

如果你想看懂什么，或者想把它记得更牢，就大声地说出来。比如:“这家伙这里写得有点深奥。我想他是这么个意思……”。更好的办法是，拉个人过来，问他“你觉得这里是个什么意思？”，然后再把你的理解讲述给他听。至于他懂了没有……反正我不太在乎。总之这样你会学得更快，这是以前光看不说的时候未曾发现的。

- **不要三心二意**

没错，这本书里面有不少好玩的图片。但别只去看那些漫画，它们是一种辅助，为了让你更轻松地理解书里面要讲的东西。这不是一本漫画书。

- **别跟你的大脑过不去**

如果你发现看的内容开始不经过脑子了，例如开始浮光掠影地浏览，不太想记住，这说明你的大脑已经有点吃不消了。这个时候你该休息一会。如果到达了某个极限时还在一味地往里面塞东西，野战排的包也受不了，更何况你 1300 克的大脑了。

- **一些小提示**

多喝点水。充足的液体供应能够让你的大脑处于最佳状态。不要在椅子或床上一呆就是两小时，多走动一下可以调整你大脑的紧张程度。此外，看完一些章节之后别再看其他书了，或者暂时不要看其他有难度的东西。你的大脑需要有一段时间对刚吸收的知识进行处理，如果此时你又往里面灌输了新的知识，那么你刚才学到的东西就会丢掉。

我相信你是最棒的。如果你对书里的内容有些疑问，或者希望更深入的探讨，给我发邮件 mess79@163.com，或者光临我的网站 http://messbook.net。我会一直等在那里。

0.4 感谢这些伙计

请允许我在这里占一点篇幅。这本书的完成与以下人是分不开的，我必须在这里告诉所有人，我以他们为骄傲：

我的父母及家人。很明显他们对人机交互一窍不通。但是他们永远都一如既往地支持着我，使我虽然离家千里却从没感到过烦恼。我非常、非常地爱他们。

编辑陈冰。感谢他对我的肯定。作为清华大学出版社最富创新精神的编辑之一，他直接影响了我的写作风格，并且对我极为信任。一个快活的老伙计。

夏德安。作为优秀的程序员、我最好的朋友之一，也是我第一本书的合著者，他一直关注着我的这次写作，并且提供了莫大的帮助——尤其是在寻找案例方面。我仍记得那些和他通宵探讨直到天明的日子（尽管其中大部分时间他都在玩 PSP）。

我的朋友曲佳。作为我的好友，以及我所在公司的 UE（用户体验）工程师，他带我参加了很多交互设计的公开活动，也认识了很多朋友。

James R Palmer。嗯，我认识的最好的历史作家，他把自己一生中最好的光阴放在了中国，研究中日战争和中俄战争。他教给我一些非常有用的写作技巧。

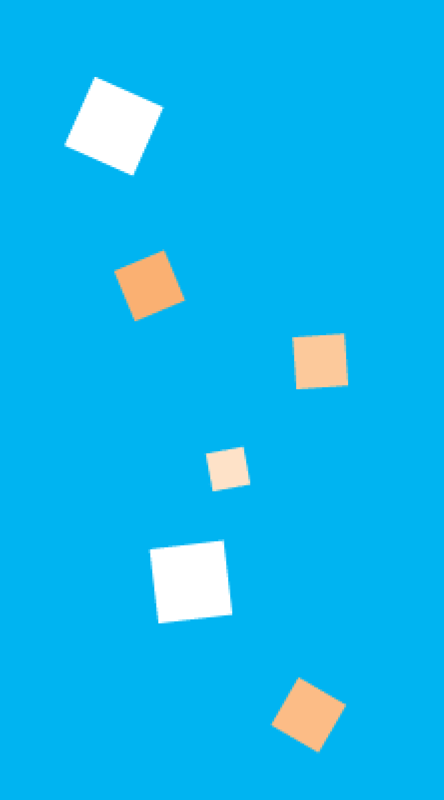
我的乐队。也许我更应该感谢他们的宽容和耐心。由于写作我失去了很多写歌和参与排练的时间，以至于无法在主音吉他手的小孩出生之前录好我们的专辑小样。非常抱歉！

向怡宁

2008 年 1 月 5 日

第一章

你首先要知道的一些事情



基督："你们必晓得真理，真理必使你们自由。"

——《约翰福音》

以下是一些我们在学习网络界面的交互设计之前先要理解的东西。有一个好的开始才能有好的结束。

1.1　我听说过这些词……不过它们到底是什么？

很多从事界面图形设计的朋友经常会彼此发牢骚："那个程序员刚才又在叫我'美工'了。"

当你问他们"那你们是什么？"的时候，他们一般都会很理直气壮地告诉你："我当然是 UI 设计师！"



我不止在一个公司发现这种情形

现在的界面图形设计行业中，"交互"、"可用性"，还有"人机工程学"之类词汇的使用频率越来越高。每个设计人员都会在解释自己设计成果的时候说："这个按钮用这种颜色、放在这里，是因为我考虑了人机工程学的一些原理。这样便于用户识别和点击。"

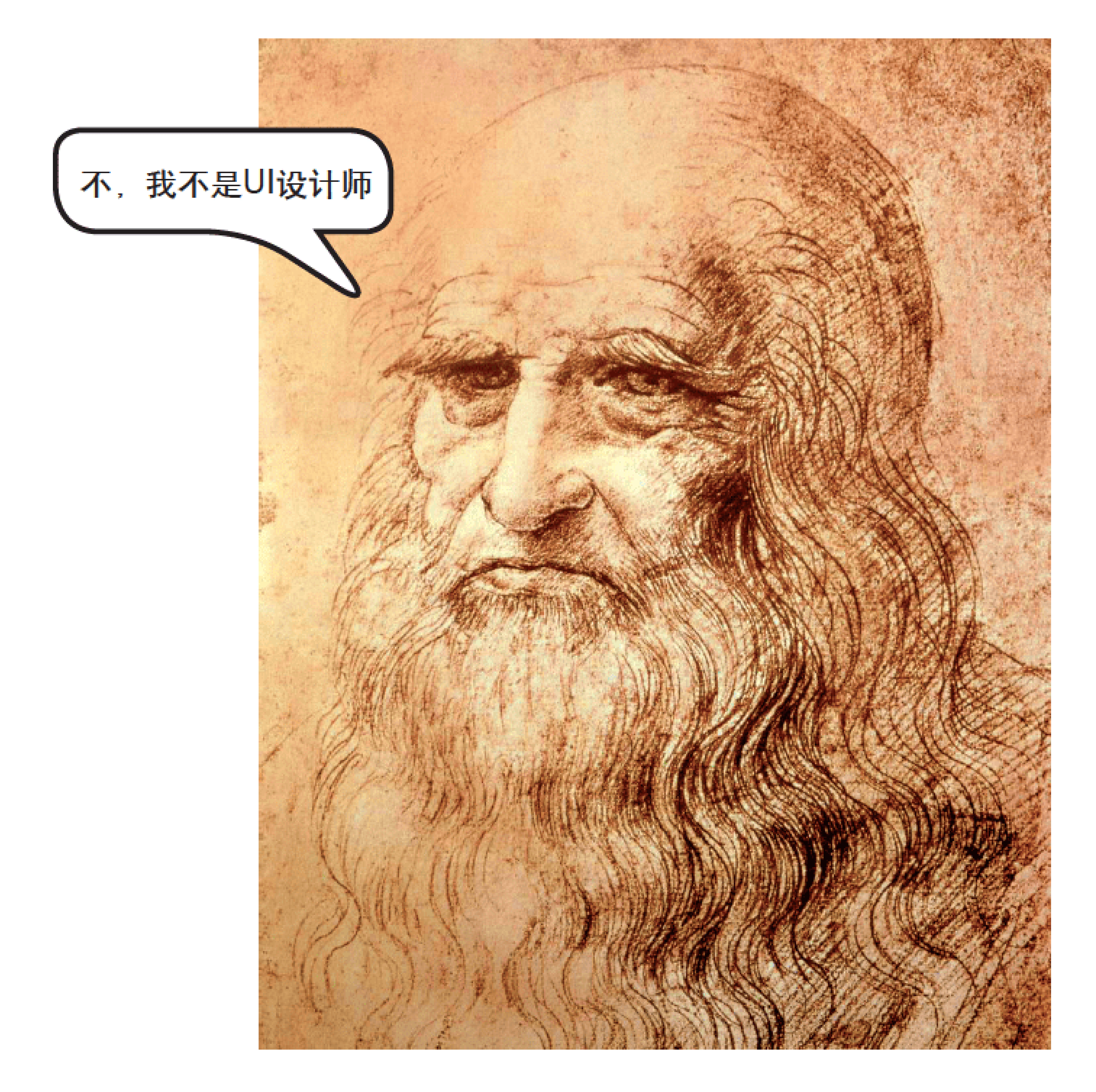
这句话说得很漂亮。而且体现出该设计人员高人一等：我不仅考虑了美观，我还考虑了更多。与此同时，这也确实涉及到人机交互学中的一些理论——比如"从用户的角度出发考虑问题"还有"突出可识别性"等等。

但实际上呢？

可能 80% 的图形设计人员这么设计和放置按钮的原因只是："我只是认为这么设计比较好，它可能会给用户带来方便。"他们都是凭借自己的设计经验得出的结论，而这些经验，并没有经过具体的考证和研究——也许它们是对的，也许不是——设计人员没有通过具体的询问用户或者

测试用户来得出这些结论。他们只是感觉这样是对的。

你可以这么做。但你不能因此就说 :“我当然是 UI 设计师。”



达·芬奇

可能很多朋友会有些生气。“美工”这个称谓当然令人感到受了些许侮辱 :它让人联想起以前工厂里的黑板报工人。因此很多新时代的设计师们对被称为“美工”非常反感。我从事了多年的图形设计，我也曾经对人说“请称呼我 UI 设计师”。但后来我了解了这两个含义之间的不同，于是稍后一段时间我对人说“请称呼我美术设计师”。直到我开始从事真正的界面设计工作（这个词的具体含义马上就会谈到），我才逐渐以 UI 设计人员自居（但仍然有些汗颜）。

那么到底什么是 UI 设计师?

1.1.1 什么是U 设计

UI 设计师，英文的全称是“User Interface Designer”，也就是“用户界面设计人员”。

用户界面设计人员的工作不是要表明网站或者软件界面完成后看起来是什么样子。他们的工作是要指出软件如何从一个版本升级到另一个版本，如何更容易地学习、使用，还有如何能帮助用户达到他们的目标。一个激情四射、吸引眼球的用户界面并不等于就是一个易学易用的用户界面。

他们研究的是产品的“可用性（Usability）”和“用户体验（User Experience）”的问题。用户界面设计，很大程度上就是在探讨如何让产品的界面更具可用性，如何让用户能有更良好的体验。这是一种优化后的界面，通过这种界面，用户能够更方便地完成任务，获得更好的感觉。

让我们打个比方。现在网页中可能要放置一个按钮。美术设计人员所需要做的，是让这个按钮更好看，显得时髦，或者符合界面中其他元素的美术风格。而用户界面设计人员所要考虑的，则是这个按钮的摆放位置、上面显示什么文字，甚至是我们到底要不要这个按钮。



很多管理者，例如产品开发经理，并不了解美术设计人员和用户界面设计人员的区别。他们认为用户界面只是一个应用程序的表面，漂亮和酷就等于好用，所以他们雇佣美术设计人员来设计用户界面。这样做的结果，往往是产品的样本演示看起来很吸引人，而一旦真正工作起来就不好用或者不实用了。

由于所谓的多媒体软件越来越受到欢迎，这种误解变得越来越普遍，审美的要求和娱乐的价值被更多地推到了首要地位。现在的公司招聘界面设计师，提出的都是关于美术设计的问题。这是一种危险的兆头，说明就连雇佣者都不明白用户界面设计究竟指的是什么。所以前面谈到的那些从事美术设计的朋友自称为 UI 设计师，也就不足为怪了。

以下是用户界面设计人员要做的各种工作中的一些例子：

- 协助为拟定开发的产品定义用户模型。
- 参与产品的可用性测试和评估，并提出修改建议。
- 将完成一个普通任务所需要的鼠标动作减少一半。
- 将网站中弹出窗口的总数量从 27 个减少到 15 个以内。
- 修改菜单中的命令，将原来不一致的、令人困惑的短语改成容易理解的词。
- ……

这些都是对“可用性”和“用户体验”的设计。它既不是美术设计，也不是写代码。感觉非常枯燥，嗯哼？

要么你就让别人去关注 UI 设计。然而一旦你完成了这些工作，或者尽可能多地做了其中的一部分工作，你的成果所带来的收益将是无法想象的。你的素质将会更高，也因此成功地迈上了新一级的台阶：你将会真正地、更为全面地看待界面设计工作，不管是美术方面、程序方面还是用户界面方面。

要想完善自己，获得更多人的承认，或者干脆承认吧，要想找到更好的工作、得到更多的薪水或者报酬，我们就必须付出一些努力。下面是需要掌握的另外一些基础知识。

1.1.2 什么是交互设计

许多产品都要求用户与之进行交互来完成它们的任务。交互就是指用户通过某种方式发出指令，例如用遥控器打开空调并降低温度，或者把微波炉设置为 3 分钟的“烧烤”模式。

影碟机、电视、卡拉 OK 里面的点歌机，这些都是交互产品。要想让产品服从你的命令，你就必须和它交互。那么如何让用户能够轻松地发出命令，或者明了产品当时的状况，就需要良好的交互设计才能解决。



这就是个交互产品……不过不够好

我们这本书主要讲的是 Web 和 Web-based 产品。举个例子来说，现在你想在网上预订一场演唱会的门票。你首先需要登录该演唱会的官方指定网站，然后选择是 VIP 席位还是最便宜的站票，再选择支付票款的方式、输入需要送票的地址，此外可能还要输入交易金额等等。这些都是交互。

你需要和某个东西（也许是网站，也许是某台机器）发生关系，给它数据或信息，然后得到它的数据或信息反馈，最终达到目的。

不幸的是，很多产品或服务在设计的时候，根本就没有考虑到用户。虽然从功能的角度来看或许可以工作，但它们往往忽略了真实用户将如何使用这个系统。

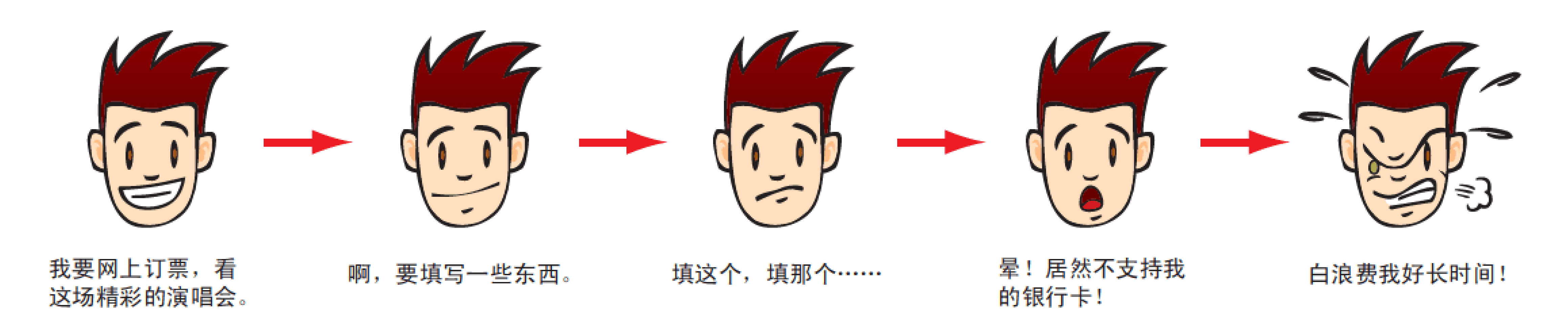
在刚才网上订票的例子中，可能会出现这种情况：你想用自己的招商银行一卡通来在线汇款。你经过了一系列订票的操作步骤，包括选择价位，然后输入自己的真实姓名、身份证号码和联系电话，填写了送票地址，甚至还选择了“希望周一至周五工作时间送票”。然而到了选择支付方式的时候你才发现，这个网站只支持工商银行的网上支付，不支持你的招商银行。

……

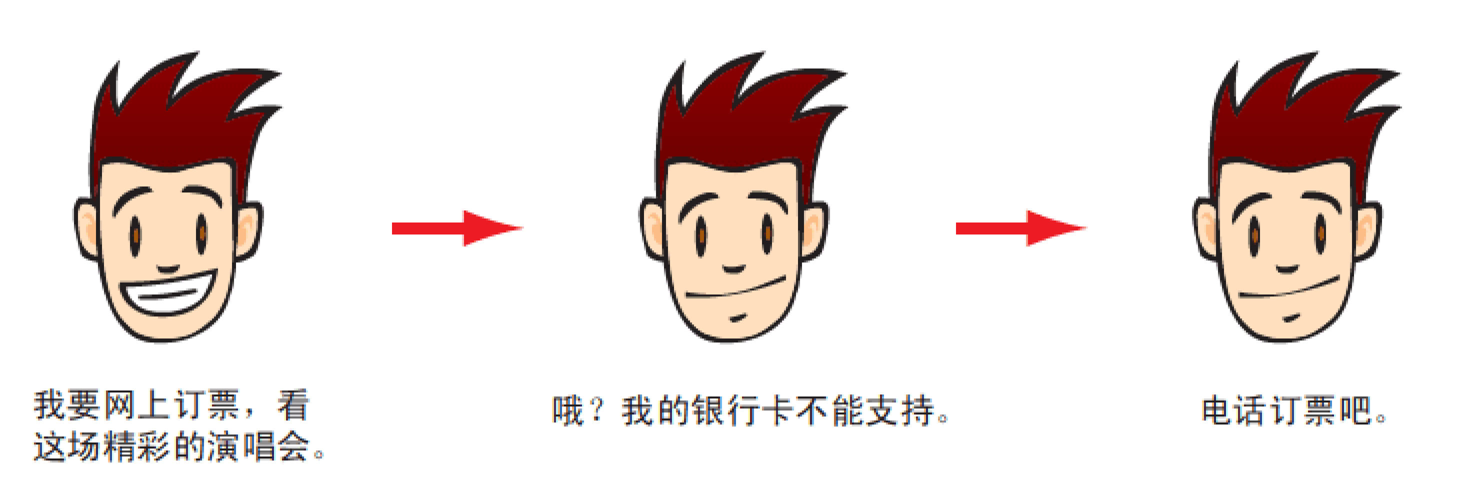
你没有工商银行的卡。你在呆滞了一分钟后关掉了订票页面，怅然若失。

为什么不在一开始就告诉我们招商银行的卡不能使用呢？或者是网站提示了这一信息但很不醒目？虽说我们还可以通过电话订票或者直接去售票点，但是已经白白耗费了数分钟在网站上去思考和输入各种信息。最重要的是，你的心情会较之刚才要糟糕一些，对这个网站的好感度则下降得更多。甚至可能你以后压根儿就不愿意再进行网上订票这种本来很方便的活动了。

你的表情大概是这样的：



没有完全考虑用户的网站



比较考虑用户的网站

而交互设计（Interaction Design）的目的，就是在设计过程中引入“可用性”和“用户体验”的概念，从而解决以上谈到的类似问题。从用户的角度而言，交互设计在本质上就是关于开发易

学易用、使用有效果而且令人愉悦的交互式产品。

所谓的“交互设计”指的是 ：

设计支持人们日常工作与生活的交互式产品。

具体地说，交互设计就是关于创建新的用户体验的问题，它的目的是增强和扩充人们工作、通信及交互的方式，使他们能更有效地进行日常工作和学习。

交互设计现在已经成了一个重要的商业领域。很多网站咨询、新生公司和移动计算行业都已经意识到交互设计在产品中起到的关键作用。对于 Web 及 Web-based 产品来说，要吸引人们的注意，就必须能从众多的竞争对手中脱颖而出，那么就面临着一个核心的问题：产品应该易用，而且有效。市场部门也开始意识到 Web 产品的可用性对于品牌的创建、点击量、客户回头率和客户满意度的影响很大。

好的交互设计可以造就一家公司的成功，反之，则可能导致一家公司的失败。因为现在有太多的网站和产品可供用户选择，如果在你这里感觉不好，他们随时可以很轻松地找到一个替代品。而当这个替代品给他们的感觉更好的时候，这些人就可能永远不再光顾你的网站了。

以前，当一家公司在计划开发交互式产品的界面时，只需招聘“界面设计师”（这个词汇在当时相当笼统），主要参与的是产品界面元素的设计。而现在，交互式产品的范围已经大大拓宽，人们对界面重要性的认识也随之提高，所以也出现了其他一些工作岗位，包括 ：

- UI 设计师（参与有关交互的各种设计，而不仅仅是界面的美术设计）
- 可用性工程师（使用可用性方法及原理进行产品优化）
- Web 设计师（设计网站的视觉效果，例如排版和按钮）
- 信息规划师（规划和组织信息的架构，尤其是网站）
- 用户体验工程师（组织用户测试、分析结果并且协助制定改进方案）
- ……

总之，用户界面的好坏不仅仅只是它看上去如何，还在于它是否易学易用、没有风险，是否能让用户在潜意识中去逐步了解它，以及它是否能更好地支持用户完成任务。

那么什么才是易学易用、没有风险的设计呢?

1.1.3 怎样的用户界面设计是不好的

我们要理解什么是好的设计，先来看看什么是不好的设计。不好的设计就是可用性不强的设计。它们一般都会包括以下几个特点（甚至更多）：

视觉表达不清

也就是说没有明显的视觉顺序和用户焦点。例如网页的内容之间没有明确划分、用户不清楚什么地方可以点击、到处都是闪动的广告和促销信息等等。在这样的界面中，用户很难把注意力集中到相关的信息上。

用户界面设计中有句话叫“用户想得到的信息才是屏幕上最应明确显示的部分”。看看下面这张图，你觉得它最明确显示的是什么（在一阵头晕目眩之后）？这种网页好用吗？



非常繁琐

也就是说，哪怕是最常见的基本任务也需要很多执行步骤。我们拿两个例子比较一下就会很清楚。

Windows Live Spaces 里面的博客系统非常简便，当你浏览某篇文章感觉有必要引用到自己博客里面的时候，直接点击下方工具栏中的“写入博客”（英文为“Blog It”）链接，便能够自动跳转到你自己的博客页面，同时文章的内容和引用地址也被自动复制到新的页面中。

在这个操作中你只需要两个步骤：

1. 点击“写入博客”。
2. 点击“发布博客”。



但我们再看看“和讯博客”的引用功能。当你看到一篇文章想要引用的时候，就比较麻烦了。你需要：

1. 先把想引用的整篇文章复制到剪贴板中；
2. 在自己的博客里面新建一个空白文章，再把复制的文章粘贴到文本编辑器中；
3. 回到引用的那篇文章，复制其引用通告的地址；
4. 点击自己的文章页面下方的“更多选项”，找到相应栏目，粘贴引用地址；
5. 点击“发布博客”。

这样的“引用”操作无疑比 Windows Live Spaces 更繁琐、不方便，没人愿意这么麻烦。因此我们可以看到和讯的博客里面很少有相互引用的，导致一个个博客成为老死不相往来的孤岛。该系统根本没有实现博客的主要功能。

提示混乱

例如没有明确告诉用户他或她所要执行的操作到底是什么意思，是否能达到自己的目标。

我们来看看 Yahoo! 中国。首页里面的搜索栏工具后面有一个叫做“全网搜索”的按钮。我不知道这个“全网搜索”究竟是指“全 Yahoo! 网站搜索”还是“全互联网搜索”？



也许有很多用户并不在乎到底是从哪里搜索得出的结果，但对我这种比较关注搜索得到的信息量大小的人而言，也许会产生一个顾虑。我可能仍然会去 Google，因为那里更明显地告诉我，我将得到的是整个互联网的搜索结果（其实 Yahoo! 这里的全网搜索同样也是全互联网搜索）。

难以使用

比如说，使用了很多用户不太能理解的词汇，或者容易引起误解的词汇。

例如某人可能会说：“这个在线银行业务要求我输入‘PIN 号’，但他们给我的卡上有个‘密码’。我不知道是不是这个？肯定是，因为他们没有给我任何叫做‘PIN 号’的东西。”但其实 PIN 是指个人身份号码（Personal Identification Number）。

又或者：“嗯。这个复选框标注着‘水平排列图标’。我想知道如果我不选中它会发生什么。我的图标会垂直排列，或者它们只是没有排列？”

强迫用户

也就是说系统提供的一些元素让用户难以接受。例如难听的按钮点击声音，或者强制播放的背景音乐。

在淘宝网，如果个人店铺的店主设置了背景音乐，在浏览时就会自动循环播放。这是个很有特色的点子。但是有一点让人很不舒服，就是当我在店铺中进行一些操作的时候，例如选择某个商品分类或者翻看商品列表，就会跳转到一个新页面，所有的元素全部重新载入。此时播放了一半的背景音乐也会中断而重新开始，尽管我其实根本没有离开过这个店铺。

另外，也许我当时正在听其他音乐，或者正在听英语，或者正在和朋友语音聊天，这时背景音乐就显得有些讨厌了。但是我把页面找了个遍也没有发现有关背景音乐的任何控件。很多店主也发现了这种情况，于是他们只能在店铺公告中多输入一些文字以提示浏览者："要想关闭背景音乐，你可以按 Esc 键。"

哦！原来我还是可以控制音乐的。但尽管这样也仍然没有完全解决问题。因为当我点击下一页的时候背景音乐又自动开始从头播放了，除非我再次按 Esc。

其实要改善这个问题非常简单：把页面改为框架结构，在主框架中提供能控制背景音乐的小工具。当子框架中页面更改时主框架不会重新载入，音乐便能够完整地循环播放了，也便于浏览者对其进行控制。



以上这些都是用户界面设计没有注意到的一些方面，也还有更多问题我没有提到。当然，这些网站在很多方面都非常优秀，但是如果能够更全面地解决可用性问题，我想它们会更加让人流连忘返。

1.1.4 那么该怎么做?

我们在设计交互式产品时，需要考虑什么人将使用它，以及它将用于何处。另一方面，我们也要理解人们在与产品交互时的各种行为。如何选择不同类型的界面、提供哪些功能选项，这些都取决于产品应支持何种行为。

什么人会使用产品？用在什么地方？

比如说现在我们有一个在线 blog 网站。它的用户范围非常广泛，可能年龄跨度也比较大，但是有一点很明确，就是用户只能够通过互联网对其进行使用。基于这一点，在开发的时候就必须注意页面下载和显示的时间，不能让页面里包含太多图片。

此外，如果这个 blog 网站主要针对年轻时尚的用户，那么可能界面风格会活跃一些；如果是面对事业有成的成功人士，那么可能体现出稳重、睿智的感觉会比较好。

用户会有些什么样的行为？

假设某个 blog 专栏首页显示的是该作者最近的 10 篇文章列表。当访问者对这位作者很感兴趣的时候，他们会想看看其他一些文章，甚至可能是很久以前写的。在这个时候，可能有些人希望找到“所有文章”这一选项便于自己选择想看的文章来读；而另一些人则可能更习惯于在打开一篇文章后在该页面直接点击“上一篇”或者“下一篇”的链接，从而直接欣赏该作者的每一篇作品。

如果该系统既有“所有文章”的选项，又在每一篇文章中提供“上一篇”或者“下一篇”的链接，那么就说明其用户界面设计在这一方面考虑了用户。

据我所知, 新浪的 blog 系统就没有这种“上一篇”或“下一篇”的功能链接。我点击某篇文章后会弹出一个新窗口，看完之后如果想要看其他文章，又会从这个窗口弹出更多的新窗口。

可能新浪在这里是这么想的：用户把首页作为出发点，每看一篇文章都弹出一个新窗口，如果想看其他的，就切换到首页窗口，打开其他文章（同样是弹出窗口）。这样首页就好像是一个出发点。但是，这个出发点并不唯一，我从弹出窗口中也能继续点其他文章。这样一来就会产生混淆：到底哪一个才是出发点？而且它也导致一个问题：如果看完文章后忘记去关闭的话，可能过了 10 分钟会出现一堆窗口, 把 Windows 最下方的任务栏都给占满了。有时候这让人很恼火。

不过新浪也并不是完全不考虑用户。文章列表中系统提供了“大中小”三个选项，是为了方便视力不好或不适应太小字体的用户，这一点是非常贴心的设计。只不过如果在前面加上“字体：”会更好一点，可以让用户更为直接地理解这三个字的含义。



我想说的是，交互设计的关键在于**如何优化用户与产品之间的交互**，从而使产品满足用户的期望（哪怕是潜在的）。在做选择的时候你应当有所依据，而这个依据就是对用户的理解。这就涉及到：

- 考虑人们擅长什么、不擅长什么。
- 考虑什么可能对用户的当前做法有所帮助。
- 考虑如何提供高质量的用户体验。
- 听取人们想要些什么，并让他们参与设计。
- 在设计过程中，让用户来尝试和验证。

1.2　我干嘛得学这些玩意……还要买本书？！

我想你已经明白了，用户界面设计并不只是“把界面设计得多么美观”那么单纯。美观是必要的，但绝不仅仅是美观就行了。没错，我们可以花上一两个星期去为新网站制作精美绝伦、眼花缭乱的 Flash 演示片头，从美术角度我们中的某个人甚至可以做到完美——但用户却可能因此花上数分钟的下载时间，也许他们在下载完毕之前就不耐烦地把网页给关掉了——连一个画面都没显示出来。

另一方面，即使产品的功能再强大，程序设计员殚精竭虑、日以继夜地开发代码，可最后如果用户下一个订单往往需要超过 20 分钟的时间，其间需要经过一系列繁琐、缓慢的步骤，填写一系列在线表格，而到头来却发现往往不能如愿，那么这个功能强大的网页或软件同样也不会受到用户欢迎。

因此，用户界面设计是非常重要的，尤其是在今天这个追求速度、简洁、方便和高效的时代。只有充分地考虑用户，以用户的感受为出发点，你才有可能制作出能经受住考验的产品——毕竟，你所做的一切都将呈现给用户，只有得到用户的承认，他们才会购买你的产品或服务，这样你的努力才不会付诸东流。

或者不妨说，只有用户满意了，你才能赚钱。



……当然这也有点夸张

1.2.1　用户界面不是次要的工作

很多人（他们主要是一些软件开发人员和产品经理）都认为，用户界面与软件开发中的其他工作相比是次要的工作。他们常常会觉得，雇一个只是偶尔才写点代码（或者干脆不写）的人来做用户界面设计和可用性测试，这简直是一种浪费。

他们的一些成见比如说：

- **认为可用性不太重要，又很浪费钱**



他们认为，用户界面的好坏对于产品在市场上是否成功影响不大。很明显这是错误的，我们之前已经举了很多不好的例子。

重视可用性问题在产品开发初期看起来似乎增加了开销（人员、时间、资金），但其实这些投资很容易就可以从不断增加的收入中弥补回来，而且会节省很多后期的开销（例如售后支持的费用）。

很容易想到的问题：一个更为可用的产品，会很快被市场和客户接纳。而如果一些产品没有考虑可用性问题而匆匆上市，这种不负责任的行为肯定会减缓销售量的增长。

- **认为用户界面不过就是“字体和颜色”**

还有些人对于“用户界面”所包含内容的理解过于狭隘。他们觉得这种工作只不过就是让界面更清晰、漂亮一点而已，认为它是软件中最为表面化的东西，可以一直等到发布之前再做，甚至干脆不做。但实际上，用户界面绝不仅仅只包括“字体和颜色”，它关心的也并不只是关于字体和颜色的处理。它关心的是软件究竟如何工作，而不是表面看上去如何。

此外，用户界面还包含一些深层次的问题。例如用户学习使用产品的时候是否感到费力、产品功能是否能满足用户的目标，用户完成任务的效率如何。这些问题和软件的标记、布局和颜色选择不同，不能在开发的后期才进行。如果不对这些问题尽早考虑，并进行测试，那么最终的版本一定不能满足要求。

- **认为用户能够适应所有的情况**

一些软件经理和开发人员坚信，只要软件提供相应的功能，用户就能学会使用。没错，人类的确有惊人的适应能力，但是认为一个潜在的用户会不顾用户界面的糟糕状况，只是因为产品的功能而购买产品，这种逻辑是错误的。

用户凭什么要这么做呢？他们可能根本就没时间去学习如何使用，也可能没有足够的动力，也可能竞争者的产品更加容易使用，也能提供相似的功能。不管什么原因，只要用户觉得你的产品“有点讨厌”，他们就不会去购买和使用你的产品。谁受到了损失？不是用户，而是你自己。

- **认为在没有足够资源的情况下可以放弃用户界面设计**

有些软件经理和开发人员在预算紧张、资源不足或时间进度要求严格的情况下，认为只能“挑重要的事情来做”，而用户界面这种“不太重要”的事情可以放在一边。这种把用户界面作为产品特性之一而牺牲掉的做法同样是不对的。


当我们有求于领导的时候……

用户界面不是产品的一种特性，甚至可以说，它超越了产品的所有特性。它可以决定每一个特性的好坏，进而决定了整个产品的好坏。一个没有有效用户界面的产品就像一个没有窗户和门的仓库：货物就在里面，但没有人能拿到。

- **认为用户界面设计只是凭空生成的灵感体现**

有些人把用户界面设计看成是创造艺术，把用户界面设计人员当成是艺人（而不是工程师），

使用右脑思索，激发灵感，然后在瞬间的火花中看到一个完美的用户界面。这一切似乎是他们凭空变戏法一般就得出了结果。

但事实上，用户界面设计工作是一种工程性的工作。以科学为基础，需要明确的要求，要考虑候选设计方案，同时它还有限制条件，需要综合考虑，还需要测试、评估和修正。这就好比程序设计也需要测试一样。不要说“我们需要你的设计一次成功”或者“为什么要测试？难道你不是个好的设计师？”之类的话，因为这是不可能的。

1.2.2　用户界面设计不是界面程序设计，也不是界面图形设计

《软件设计的艺术》的作者，Terry Winograd（2004 年度的 ACM 人机交互院士）曾把交互设计和软件工程比作是“建筑师与土木工程师之间的区别”。我们不妨对此观点稍加引申。

可以说，交互设计、美术设计和软件设计这三者的区别，有点类似建筑行业中的建筑设计师、装修设计师和土木工程师。

- **建筑设计师最关心的是房屋居住者的感受。**这些问题包括：房屋的外观是不是足够好看？客厅会不会太大、餐厅会不会太小？卫生间是一个还是两个？居住者会不会觉得现在的房屋空间不尽如人意？

- **而装修设计师则考虑的是在现有的空间中如何美化环境。**这些问题包括：房间的整体色调是明快还是稳重？客厅中需不需要加上修饰作用的背景墙？地板是深色还是浅色？地毯、沙发以及茶几之间的造型、色彩搭配应该如何考虑？

- **相比之下，土木工程师则更关心工程的实现。**这包括一些实际问题，例如成本、工期和结构问题、环境问题、防火措施以及建筑方法等等。

可见，“设计”、“修饰”和“建造”房屋彼此之间并不相同。与此相类似，交互设计、美术设计和软件设计之间也存在着差别。



软件设计师可能存在的问题

学会了如何识谱，不一定就能写出脍炙人口的乐曲；会做木匠活，也不能说明就能设计出别具一格的家具。同样的，知道如何使用编程工具和编程组件，也不代表你就一定能知道如何创建可用、好用的软件。

即使是拔尖的程序员，对开发工具非常的了解，也仍然可能开发出拙劣的用户界面，即便是在时间和资源都很充足的情况下。导致这种现象发生的原因有几个：

- **缺乏设计用户界面的经验**

  一个编程经验丰富的程序员并不一定具有很多设计用户界面的经验。与此类似，进行用户界面设计也并不一定需要太多编程的知识。



- **程序员对于某个工具越熟悉，就越容易受其影响**

  他知道这个工具的能力，他驾轻就熟。但工具提供的功能并不总是与用户的需求完全吻合。

- **拔尖的程序员往往是一些意志坚强的人**

  他们很牛，也知道自己很牛。所以他们不愿意和他人协商，也不会轻易妥协。如果一个开发队伍里面有几个这样的成员，而同时他们的管理者没有足够的权威，那么结果往往是不同程序员开发的不同部分之间存在着严重的不统一，每个人都以自己的方式去做自己的事情。

美术设计师可能存在的问题

美术设计人员更擅长于设计各种元素和控件的艺术外观，形象地表达功能，为整个应用程序创建一致的艺术风格，同时让那些数据排列起来更加好看，让用户感觉这个网站或产品确实很漂亮或很酷。

但是，尽管美术设计师们有着良好的美学素养，他们并不一定就适于用户界面设计。他们很可能只去在乎如何让界面更酷更炫，但是却不会去关心怎样能使用户更方便地使用产品，他们也不知道怎么做能让用户更方便。漂亮和酷并不等于好用，如果只去考虑“外观美不美好”，很可能最后产品只是“看上去很美”，但其实只是个绣花枕头。

记住，设计用户界面和设计用户界面的图片是不同的行为。

真正的用户界面设计人员

这些家伙擅长的是分析和理解用户的任务需求，使控件和信息的安排更加合理，还有简化产品的复杂度。同时他们能认识到用户在学习或使用产品的过程中，一般可能会在哪些地方遇到问题。

任务分析、概念设计
提供实时响应的标准
制定一致性的标准
可用性测试和评估
……



用户界面设计师

编程、内部结构
使用界面实现工具
重视功能的实现
理解技术局限
……

软件设计师

页面布局
设计图像或标志
艺术的外观设计
形象地表达功能
视觉上的一致性
……



美术设计师

一般来说，应该由用户界面设计师来设计界面，然后美术设计师对其进行艺术加工，最后软件设计师来编写程序将其实现。这是一种分工合作的产物。

当然，作为一个开发团队，任何角色的成员都可以来参与用户界面设计和评估，发表看法、交换意见（我非常鼓励这么干）。但是用户界面的设计主导工作，仍然需要专业的用户界面设计师来完成。

1.2.3 交互设计是一门跨学科领域

现在的世界是个什么样子？语音识别、多媒体、移动计算技术、网络技术、可视化信息……这些新技术的发展浪潮一波接一波，人们总是在不停地展望和发现新的学习、交流、工作和生活的方式。与此同时他们还把视线转向了人生的方方面面——不论是在家里，还是在学校；不论是在休闲，还是在工作中；也不论是独处，还是与家人或朋友共度——所有这些细节都在启发新的设计构思，从而不断地增强和扩充原有的交互方式。

所以，交互设计的范围也越来越广大，与各种学科相互交叉渗透，共同发展，相互之间的要求也更高。

简单一点说，与交互设计相关的学科包括：

人类工程学	人类感知科学	心理学	软件工程学
信息处理学	市场行为学	工程学	行为学
计算机科学	社会科学	人类学	工业设计
图形设计	产品设计	美术设计	信息系统

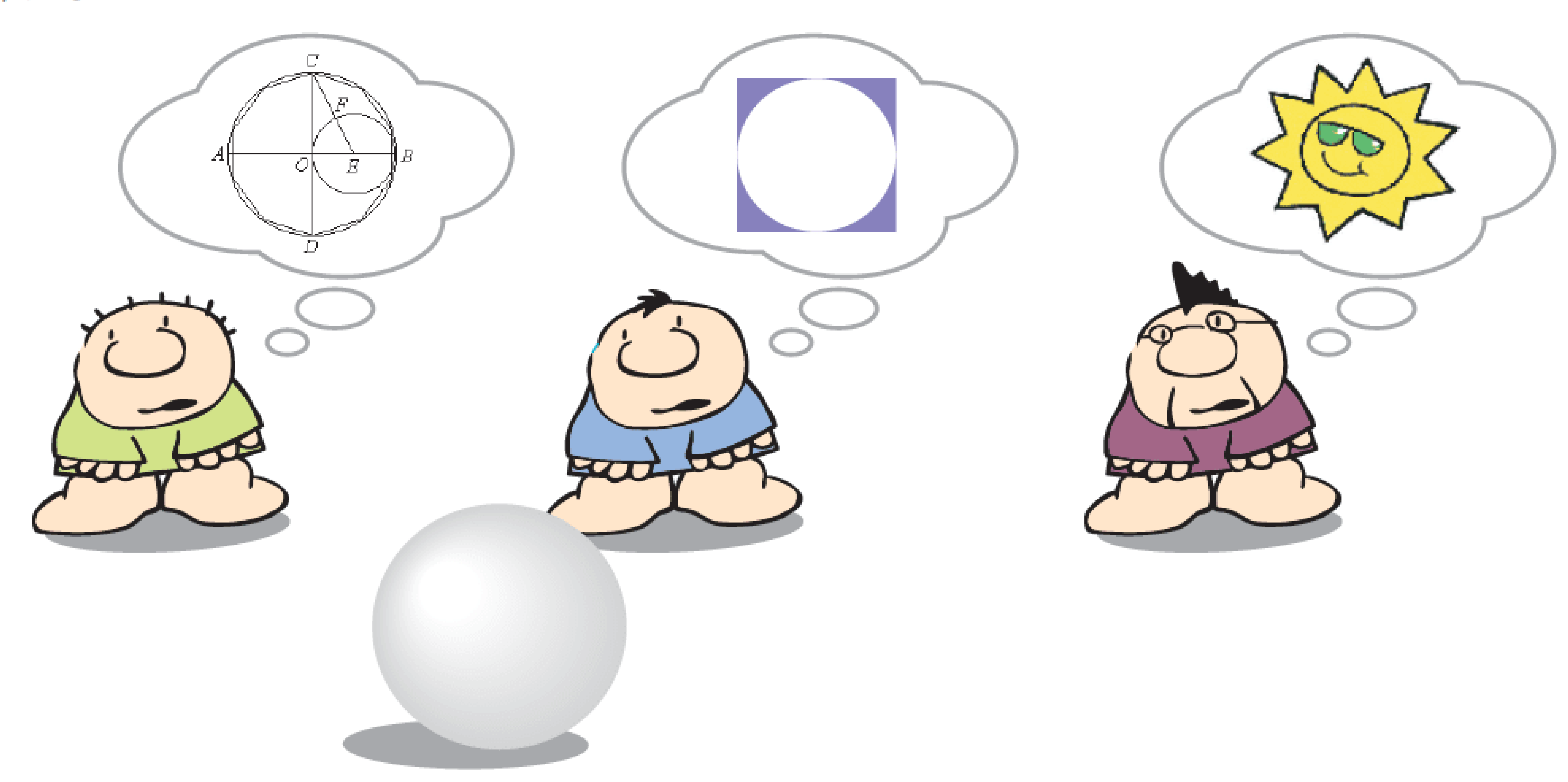
举一个浅显的例子。如果我们在开发一个供儿童使用的交互式讲故事网站，就需要知道怎么给儿童编写故事，他们才会觉得精彩。这涉及到人类心理学的知识。而怎样讲他们才能理解，就涉及到人类感知科学。当然还有网站的按钮怎么摆放、有没有背景音乐和动画，这些又牵扯到人类工程学和行为学。此外网站的制作又需要软件工程学和计算机科学、美术设计等等。

更别提那种艺术展览的交互式介绍指南了，你还需要对艺术作品有很深的理解能力，还要知道人们的欣赏习惯和游览路线。

这些都需要很专业的知识，就好像不懂油盐酱醋，就做不出好菜。为了解决这种情况带来的问题，人们也想出一种办法，那就是跨学科小组协作。也就是把许多有着不同背景、受过不同专业训练的人员结合起来，共同参与产品设计。这样将会带来更多的构思，有助于开发新方法和提出更新颖、更富创意的设计。

但另一方面，这种协作也会带来很多问题。首先就是成本。如果是各行业的专家和带头人，那么成本会更高。谁也不知道这种大规模的邀请会花多少钱。

其次，背景不同的人员越多，成员之间的沟通和设计的进展也就越困难。因为不同背景的人，在观察和讨论同一个事物时会有不同的视角和方法。某个人认为重要的东西，其他人可能根本就视而不见。

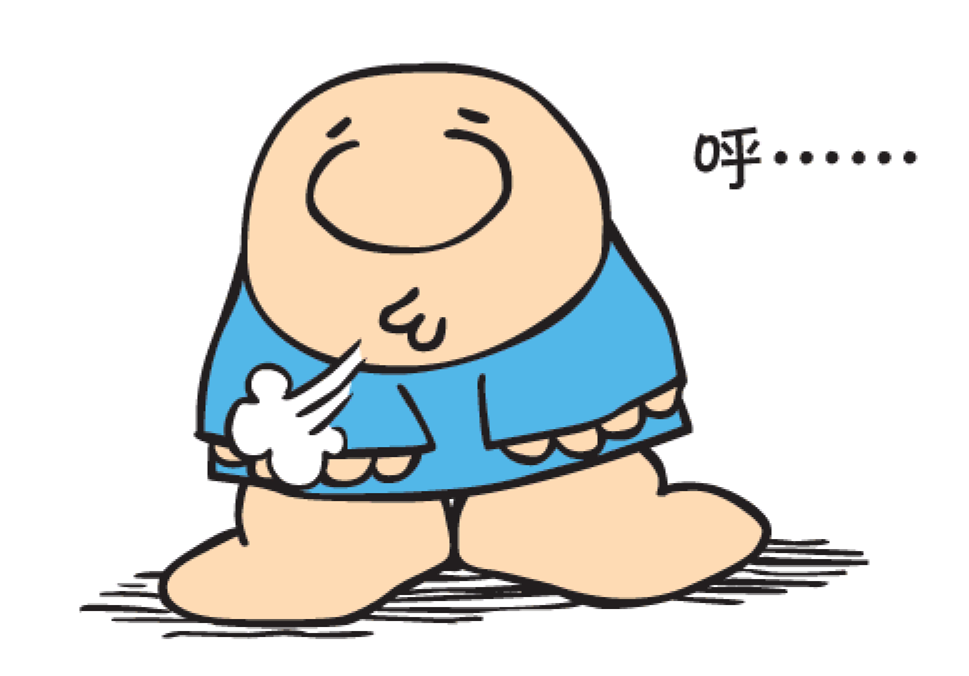


也就是说，如果是跨学科小组协作，那么小组中可能经常会出现混淆、误解和沟通障碍的问题。这有点类似我们常说的“公说公有理，婆说婆有理”。如果设计组的成员没有什么团队协作的经验，可能问题还会更多。

似乎这是一条不归之路……那么该怎么办呢？

1.2.4　不用成为专家，理解方式即可

放轻松一点。没有人是全才，也没有谁要求你精通一切。



正因为我们不可能成为涵括上述那些领域的专家，所以才会有人总结出多年的经验和成果，提出了一系列交互设计的科学方法。这些科学方法是在积累了许多人的智慧、汇集了几十年交互系统设计的经验基础上建立的，也是建立在关于人类学习、认知、阅读和理解的一个世纪的研究基础之上的。

通过这些科学方法，你就能按部就班地开展工作。这有点类似“依葫芦画瓢”，但同样也需要付出。

从另一方面来说，也没必要去成为交互设计或者人机交互设计的专家。没人逼着你考级、评职称。你的目的只是为了让自己的产品或服务能够变得更好。也许做着做着你就能成为专家。

所谓多少年总结出来的人机交互设计的科学方法，其实就是下面这句话：

定义用户，以用户为中心进行设计。同时不断地进行用户测试并对产品进行修改。

仅此而已。

在网页或基于网页的软件设计中，我们时刻把握住这句话，不需要专家们的介入，我们自己能够完成一切。

Google 那句话怎么说来着？

“Focus on the user, and all else will follow.” —— “以用户为中心，其他一切纷至沓来。”



1.3 OK，那我该怎么做？

我想你已经了解了交互设计的重要性。简而言之，用户界面设计就是在产品功能完整的基础之上，考虑如何更好地展现，以及如何让用户能够更方便容易地使用它，从而在各个方面提升产品的质量，得到用户和客户们更高的评价，最终扩大产品和服务的销售。

从我们自身来说，好的交互设计能够更充分地体现开发人员的价值。作为一个 UI 设计师，做出成绩是自身得到更好发展的前提。

很多公司意识到交互设计的重要性。但同样有很多公司仍然不知道该如何进行。那么到底该怎么做呢？

首先我们需要知道有关交互设计的具体组成以及涉及的各种过程。全面地了解交互设计，摸清它的轮廓，我们才能明白这到底是怎么回事，在这些过程中我们需要做些什么。

1.3.1 交互设计的4个内容

一般来说，交互设计过程涉及以下 4 项基本活动 ：

理解用户的需要，从而建立明确的需求

开发满足需求的一些候选设计方案

制作设计方案的原型

用户测试和评估

这些过程的目的，是要保证产品是可用的。因为不同的用户有不同的需要，你需要相应地设计不同类型的产品。例如，儿童和成人之间的不同，会导致他们的衣服、食品和游戏都各不相同。做做小游戏、加入点卡通形象更有助于激发儿童的学习兴趣，但这种方式是否适用于 25 岁的男性研究生就很难说了。

为用户“度身定造”产品或服务，这一点通常可以采用**“以用户为中心”的设计方法**（也就是 UCD 方法，User Centered Design，记住这几个字母）来实现。“以用户为中心”，也就是要一切以用户的最终感受为出发点，让用户参与整个设计过程。同时，为了验证产品确实能够满足用户，你也需要通过测试和评估来进行判断。

下面我们再来看看交互设计的 4 个基本活动。

理解用户需要，建立用户需求

我们必须了解谁是目标用户、他们需要哪些支持，然后才可能设计出能支持这些用户的产品。首先要找到真正的用户，然后从他们那里找到需求。用户和他们的需求是所有设计、开发工作的基础。在“以用户为中心”的方法中，这个活动是最基本的，“万物之源”。简而言之，在这个阶段我们需要知道：**1. 用户想要什么；2. 我们要做什么。**

开发一些候选设计方案

这是设计的核心活动，也就是提出满足用户需求的构思。它可以划分为两个子活动：概念设计和物理设计。所谓概念设计就是用一种用户能够理解的方式去描述产品应该做些什么、如何运作、外观如何；而物理设计考虑的则是产品的细节，包括要使用的色彩、声音和图像，还有菜单设计和图标设计。

这些设计构思都需要提出各种候选方案，从而制作出原型以供评估。这个阶段其实就是：**试试该如何来做。**

制作设计方案的原型

评价一个交互设计的最佳方法，就是让用户与产品进行交互，让他们去感受。这就要求我们要把设计方案转化为可以交互的版本，或者说产品的原型。

这些原型并不一定必须是能够运行的真正软件版本，我们可以采用很多技术来实现所谓的“交互”。比如说，制作纸张原型（把步骤画在纸片上并口头为用户演示）就又快捷又便宜，而且能在设计的早期阶段就发现问题。这个阶段概括来说就是：**先简单地做个样子出来。**

用户测试和评估

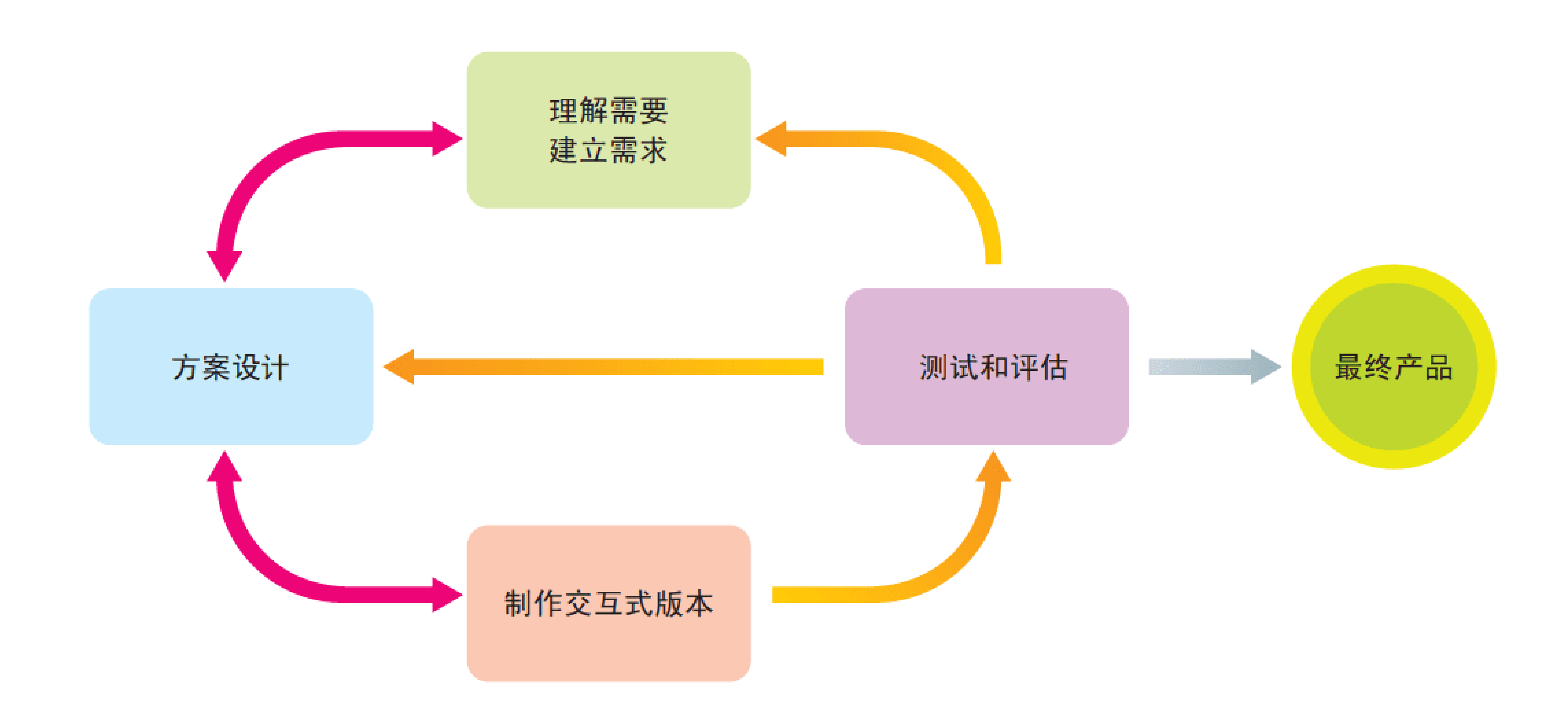
也许你和你的同事都能自如地浏览并使用自己开发的网站，或者能够顺利地通过产品或软件完成某个任务。你们甚至还征求了专家的意见。你们可能觉得不需要其他什么保证了。

但是适应开发者和专家的产品并不一定适合绝大多数普通用户。作为开发者，你对自己产品了解得太多了，你清楚网站里栏目的内容、产品的层次和架构，你也知道要想完成一个任务应该按什么顺序点击什么按钮。而且作为自己领域的专家，你了解每一种流行的趋势和约定俗成的使用规则，熟知每一个专业词汇和术语，但其实这些规则或词汇很可能让那些真正的使用者感觉根本不知所云。

这就好比要想知道设计的玩具是否能让小孩感到好玩，你就需要把玩具塞到他们手里看看他们的反应一样。我们需要让用户自己来检验。同样，用一句简单的话来概括这个阶段也就是：**看看这个设计行不行。**

这些活动是相互联系的，交织在一起，也需要重复进行。比如说，制作了某个设计方案的原型后，我们需要让用户来进行评估，从而得到一些反馈信息，如必须进行哪些修改，或者哪些需求仍未满足等等。这样就要求我们再次找到用户明确需求，然后提出修改设计的方案。同样的，这些方案仍然需要测试和评估以获得新的反馈。

这个相互联系并且重复进行的特征被称为“迭代”，是交互设计过程的关键特征之一。



1.3.2 交互设计的3个特征

那么我们就来看看交互设计过程的 3 个关键特征：

以用户为中心

确定具体的可用性标准，并明确建档

必须迭代进行4种设计活动

同样的，我一个一个地向你解释一下。

以用户为中心

“以用户为中心”的意思就是“让用户参与整个项目开发过程”。这是交互设计的一个重要观点，因为只有用户才是产品成功与否的最终评判者。产品只有在用户满意的条件下才可能卖得好，从而为你带来效益。用户不满意的产品肯定会被淘汰。

也许我们并不能保证让用户参与开发的方方面面，但把“以用户为中心”作为一个口号，能鼓励我们关注这些问题，并提供让用户参与评估并提出反馈的机会。

建立明确具体的可用性标准

在项目开始的时候，你就应该确定这个项目的可用性目标和用户体验目标，并做出明确说明，同时所有的开发人员也应该就这些问题达成一致。这些标准是后续工作的纲领性文件，一旦明确建档，任何人不得有异议。这些标准便于你选择不同的候选方案，而且在产品开发过程中，你也能随时根据标准检查这些方案。有这么一个纲领文件，在设计了一半的时候如果对某些根本问题出现了争执，你也好办得多。

反复迭代

“迭代”就是反复进行设计过程中的某些阶段。通过“迭代”，你就能利用反馈来改进设计。反复进行“设计—测试”，测试就会更全面、更彻底，最终能够发现并解决绝大多数的问题。

另外，由于用户和设计人员都参与设计，也许有的人希望这种方案，而另一些人则喜欢那种。这也需要进行“迭代”，检测方案是不是能够满足需求。而且各个方案也能相互启发，取长补短。不论设计人员多么优秀，也不论用户对新产品多么有想象力，都必须使用反馈来修正构思，而且需要反复若干次。在设计创新的产品时，这一点尤其重要。

迭代是不可避免的，因为设计人员不可能一次就找出正确的解决方案。你能够保证你的每次决定都万无一失么？

1.3.3 交互设计的两个目标

没有明确的目标，做事情当然是无方向性的，往往事倍而功半。

产品的功能目标是满足用户的使用需求。例如 Excel 就必须能对表格的内容进行运算。而交互设计的目标，则是让用户在完成任务或学习的时候更方便、更舒心和更有效，甚至是更有乐趣。

这个问题很容易想象。一个管理系统，自然要非常有效，从而能够提高用户的工作效率；而一个学习软件，则需要表现得更有挑战性和吸引力，让用户能够更积极地学习。这些“高层”的关注事项我们称之为交互设计的目标。

这些目标可以具体分为“可用性目标”和“用户体验目标”两种。可用性目标是关于满足特定的可用性标准（例如有效性）的，而用户体验目标则是对用户体验质量所做的明确说明（例如富有美感、令人舒畅）。

可用性目标

所谓可用性，就是说产品是否易学、使用是否有效果，以及通用性是否良好等等。它涉及优化人们与产品的交互方式，从而使人们能更有效地进行日常工作、完成任务和学习。

可用性目标具体可以分为：

- 可行性（使用有效果）
- 有效性（工作效率高）
- 安全性（能安全使用）
- 通用性（具备良好的通用性）
- 易学性（易于学习）
- 易记性（使用方法便于记忆）

我们分别来看看这些可用性目标都是指什么。

- **可行性**

可行性是最常见、最基本的目标，指的是产品是否“可行”。也就是用户能否通过产品达到意图，还有达到意图的程度有多少。很简单的道理，如果产品或服务使用起来没有效果，谁还会用它呢？

也就是说：产品是不是允许人们轻松地学习、有效地完成任务、访问所需的信息，或者购买所

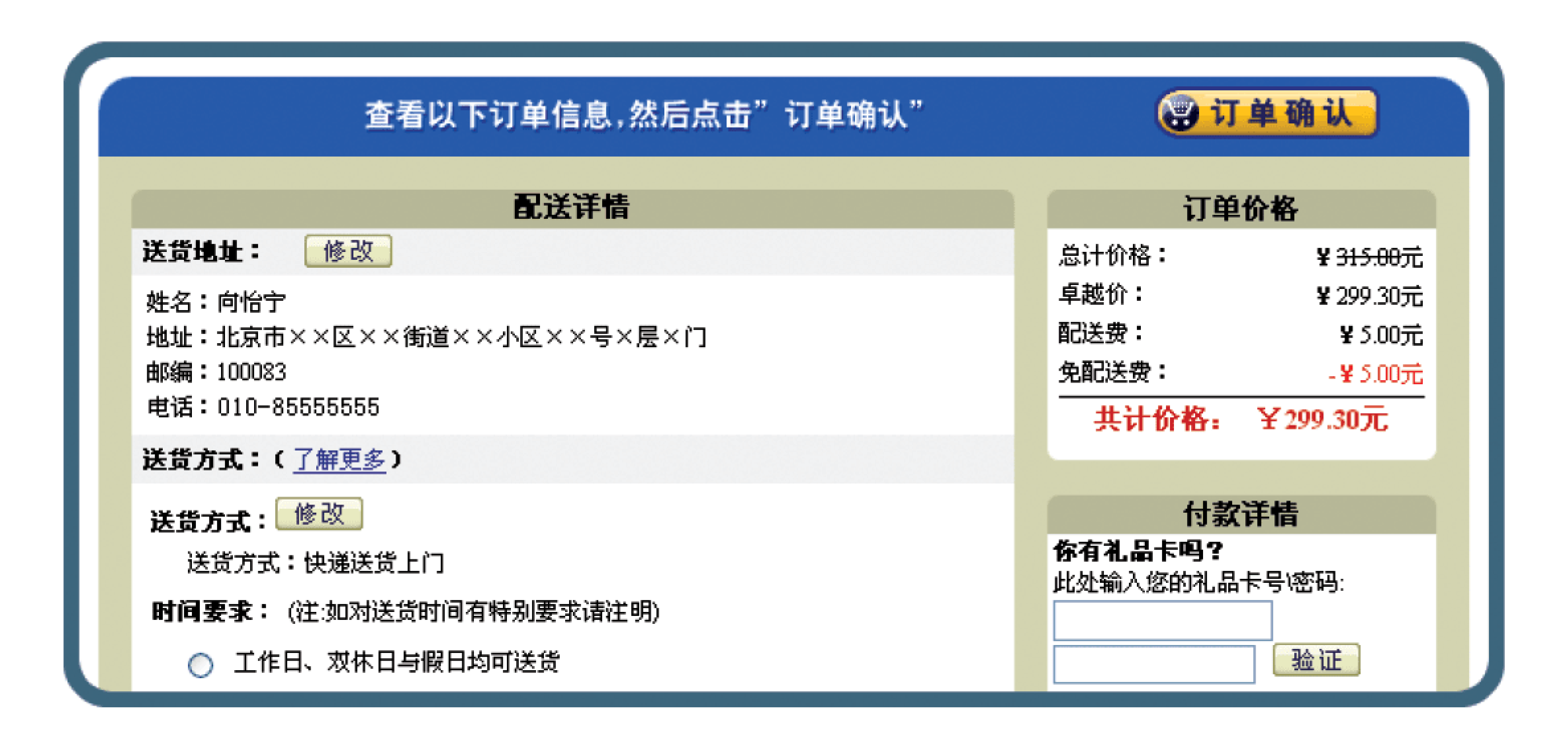
需要的物品？

● 有效性

有效性指的是用户在执行任务时，产品支持用户的方式是否有效，从而避免繁琐的操作。

当我们第一次在“卓越亚马逊”进行在线购书的时候往往需要注册。注册可以输入购买图书后的配送地址，网站将会对其进行保存。这样，当我们下次在“卓越亚马逊”买书时，网站会自动将保存的配送地址显示出来，无须再次输入。要知道，输入“××省××市××区××街道××号×层×门”确实需要一些时间，而且大部分人在这个过程中都有些烦躁。

当然，你也可以随时修改配送地址。



在这里我们要问的是：用户在学会了如何使用产品完成任务之后，能不能维持一个较高的工作效率？

● 安全性

安全性关系到保护用户以避免发生错误，以及令人不快的情形。不管是新手还是老手，他们都有可能会犯错误。产品应该能避免因为他们偶然的活动或误操作而造成损失。要知道，每个人对出错都会担忧后果，这种担忧对他们后续的操作会有很大的影响。比如说你需要：

■ 降低用户按错键或者按钮的风险，从而预防用户犯严重的错误（例如不要把“退出”或“删除”命令与“保存”命令安排在一起）；

■ 为用户提供出错后的复原方法，从而让他们更有信心，而且敢于发掘界面、尝试新操作；

■ 采用其他安全机制，包括取消选项和确认对话框，让他们有机会再次考虑自己的意图。

例如网易的电子邮箱系统，当你选定了要删除的信件时，网页会产生一个对话框并询问“是否确定？”，从而防止你误删除了重要的邮件。



我们对此需要知道：系统能不能预防用户犯严重的错误？如果用户已经犯错，好不好恢复？

● 通用性

通用性指的是产品是否提供了正确的功能接口，以便用户可以做他们需要做的、或是想要做的事情。

如果有这样一个绘图软件，它只能使用鼠标而不支持手写板，而且只能绘制多边形，那么它的通用性就很差，没有多少人会使用这种“讨厌”的软件。

也就是说：系统是否提供了适当的功能，使得用户能够以适合自己的方式来完成任务？

● 易学性

易学性指的是学习使用产品的难易。对任何产品，用户都希望能立即开始，而且不费多大力气。经常使用的产品更应该这样。我记得有一个标准叫做“10 分钟法则”：“简单常用的系统，新用户应能在 10 分钟内学会如何使用它”。

也许用户愿意花较长的时间去学习复杂的、提供更多功能的产品，例如 Dreamweaver 和 Photoshop。但对于 Web 产品来说，用户更急躁，要求更高。就算有在线帮助，通常也很冗长乏味，而且很难与实际任务相联系，所以没有多少人真正去使用。

因此，一个很关键的问题是，你得确定用户到底愿意花多少时间去学习这个系统。如果大多数用户无法或者不愿花时间去学习，那么去开发它还有多大必要呢？

我们对此需要知道:1. 由开始使用产品到能够执行经常性的任务,需要多长时间?会不会很难?2. 学习了各项功能后,完成更复杂的任务,需要多长时间?会不会很难?

- **易记性**

易记性指的是用户在学会某个产品后,是不是能迅速地回想起使用方法。这一点对于偶尔才使用的交互产品尤为重要。用户们不应该每次都需要重新学习如何执行任务,起码借助一些简单提示就能回想起它的用法。

如果产品的操作含糊、不合逻辑,或者次序不合理,它的使用方法就可能很难记住,用户会经常觉得需要帮助。

有很多方法可以提高易记性。例如在执行任务的不同阶段,使用一些有意义的图标、名称或选项来协助用户记住操作次序。另外在组织选项和图标时,把它们进行分组(例如把所有绘图工具放置在界面的同一个区域),也能使用户知道在什么阶段应该在哪里寻找这些工具。



这里的问题是:产品提供了哪些类型的界面支持,用以协助用户记住如何执行任务?

以上这些可用性目标非常适合于 Web 和 Web-based 产品。它们的目的是要通过改进方式,从而提高效率。把这些可用性目标表达成特定问题的形式,它们就能转变成可用性标准。而通过这些标准,我们就能衡量某个产品是不是具有可用性。

用户体验目标

随着新技术的快速发展,人们对产品也有了更多的要求。这就使得研究人员和业界人士开始思考进一步的目标。交互设计已经不仅仅只是如何提高工作效率的问题了,人们也越来越关心产品是否具备其他一些品质,例如:

- 令人感到满意
- 令人心情愉快
- 有趣味性
- 引人入胜
- 富有启发性

- 富有美感
- 富有时尚感
- 可激发创造性
- 让人有成就感
- 让人得到情感上的满足

所谓的“用户体验”指的就是用户在与系统交互时的感觉怎么样。比如说，“动感地带”网站的主要消费群体是年轻人，它的主要目标就是有趣味性、富有时尚感。年轻群体更喜欢夸张、动感、戏谑，他们不仅仅只需要网站的功能。



用户体验目标与可用性目标不同。可用性的目标非常客观，而用户体验目标关心的则是用户自己的情绪和感受。用户体验并不是评价产品多有用或者多有效。可以说，可用性目标是交互设计的核心，它采用的是明确的衡量标准。而用户体验目标的定义则不那么明确，因为不同用户的感受是不一样的，这个没有办法来加以量化。

当然，可用性目标和用户体验目标之间也存在着一个权衡折衷的问题。我们所要注意的，是在满足用户需要的前提下，追求二者的不同组合会对产品有什么影响。并不是所有的可用性目标和用户体验目标都适用于每个产品，有些目标甚至是相互排斥的。例如你没必要去设计一个既有

趣又安全的网络管理系统。还有的设计师为了界面更为美观而运用大量的色彩，结果却令人很难找到自己感兴趣的东西。

你问我哪些目标最重要、哪些次要？我想我没办法回答这个问题。答案取决于使用的环境、具体的任务以及针对的用户。

比如说，如果是一个供设计师使用的绘图软件，就需要易学、易记、具备良好的通用性，同时安全、有效，能够激发创造性并富有美感。

而一个以恋爱、婚姻、感情为主题的札记网站（例如“情感天地”），就应该易学易用、富有启发性，同时能让人得到情感上的满足。

1.3.4　摩西的十诫

既然它们是目标，我们就应当时刻注意要满足这些目标。

把可用性目标概念化，就会得到一些可用性原则。这些原则是由理论知识、实践经验和尝试结合而成的，它不会告诉你如何设计实际界面，而是一组提示，是为了帮助你解释和改进设计，而且确保在界面上提供了必要的内容。

我这里有一系列针对所有交互产品的可用性原则。它们有十项，你不妨把它们看作是交互设计的“摩西十诫”。

让用户随时了解系统的状态

在适当的时候应该提供适当的反馈，让用户能随时掌握系统的运行状态。反馈就是要返回与活动相关的信息，例如你已经执行了什么动作，我现在正在帮你上传文件等等，这样用户就能够踏实地继续他的操作。

如果某位用户做了某个操作，却得不到信息，或者要等很长时间才知道他所做的操作是否确实有效果，那么他的效率和心情可想而知。



系统应与真实世界相符合

你应当使用用户的语言。也就是使用他们熟悉的那些词汇、惯用语和概念，而不是高深莫测的技术用语。

此外，界面的设计应当遵循一些公认的惯例，体现出控制与效果之间的良好对应关系。

比如说，在现实生活中，录音机的倒带键在左、播放键在中间，而快进键在右边。这种安排和磁带的运动方向是吻合的。如果颠倒了按键次序，很可能就难以使用。这个惯例一直影响到 CD 机甚至 MP3 的按键设置（哪怕它们并没有具体的运动方向）。

在计算机中的媒体播放软件同样都遵循这一原则。例如下面这两幅图。上图是自然映射，往左是“上一曲目”。下面那个就有点别扭了。



另外，界面还应当具有“启示性”，告诉用户对某个元素能够采取什么行动。例如按钮的设计应暗示着这个玩意可以点击，滚动条要体现出它能够被上下移动等等。

给予用户控制权和自主权

人们对自己无法掌控的东西总是有些敬而远之。不管是网站还是其他产品或服务，用户总是主角，是他们控制着系统。但是，系统毕竟是死的，它不能像人类那样能够给我道个歉或者问我一句“怎么办”。这种时候你就必须想点办法来弥补这一缺陷。

例如，当用户在操作中发生了异常状况时，你应该为他们提供标记醒目的“紧急出口”。也就是说，让用户有权自主中断某项进程，而不是当系统出错或误操作后只能看着错误继续发展而无可奈何。

想想，如果你在某个网站上传视频，结果在开始上传后发现它是个，咳……有点影响社会风气的东西，却发现不能取消，是不是就有点坐不住了？



提倡一致性和标准化

一致性指的是在设计界面时使用相似的操作，并且为相似的任务使用相似的元素。这就好像在你家厨房和洗手间里面，冷热水龙头的左右位置总是一致的。如果厨房是左边出热水，洗手间里面则是右边出热水，你可能就会经常弄错。

一般我们都是点击鼠标左键来选取界面上的对象。而如果有个界面，某些对象是用左键选取，还有一些对象却只能够用鼠标右键选取，这种相当随意的不一致性就会导致用户很难记住，当然也会更容易出错。

另外，如果用户无法确定不同的词汇（或者情景、动作）是否具有相同的含义，他们也会很犯怵。比方说你的网站里面确定按钮都是“确定”两个字，但是突然冒出个“OK”或者“好的”，用户就会很莫名其妙。

帮助用户识别、诊断和修复错误

当出现错误时，使用简明的语言描述问题，并且能够推荐解决方法，而不是冷冰冰地出现一行文字：“由于类型 2 错误，本程序意外终止”。

Gmail 邮箱起码能告诉我大概的问题原因和解决办法：



预防错误

比帮助用户修复错误更好的，是在一开始就预防错误发生。

在某些时候你可以合理地限制用户的操作。比如把某些按钮或菜单选项设置为灰色让它不能点击，从而把用户的活动限制在允许的范围内。他们无法进行不正确的操作，自然错误就不太可能发生。



依赖识别而不是记忆

我们需要使对象、动作和选项都清晰可见，而且标识性强。例如，使用简洁明了的图形或文字提示用户某项操作的功能、目的以及可能发生的结果；或者把同类型的命令或选项归置在同一个区域内，这些设计都能帮助用户在不自觉中加深印象。

对于网站来说，如果导航太复杂、选项太多，也不方便浏览者识别，记忆就更难了。比如说下面这个导航：



强调使用的灵活性及有效性

比如说，你可以提供一些新用户不可见的快捷键，这样既不会干扰新用户，又能方便有经验的常用用户能快速地执行任务。

当然并不是指新用户就不能使用快捷键。最简单的例子：当我们在搜索框里面输入关键字后，应该按“回车”键就能触发系统开始搜索。如果此时还要把手从键盘挪到鼠标上,再去寻找“搜索”按钮点击，很明显就不够灵活有效。

最小化设计

避免使用无关或极少使用的信息。尤其是网站设计。

要想最小化设计，一个比较好的方法是：把每个你认为可能不必要的元素逐一去掉。如果没有它系统仍能很好的工作，那么就撤销它。

如果把过多的信息塞进有限的屏幕空间中去，用户便很难找到自己感兴趣的东西。撤除那些不会影响网站或软件总体功能的元素总是有益的，可以使网站或软件更为简洁、明快和易于浏览。

当然，一定数量的图形、阴影、色彩和格式能够增加视觉上的美感，对于用户也是一种享受。没有任何图片的网站肯定没有多大吸引力（想想一个没有图片的 Cartoon Network）。所以，设计的关键是要在网站外观的吸引力与内容的数量、类型之间取得适当的平衡。

提供帮助及文档

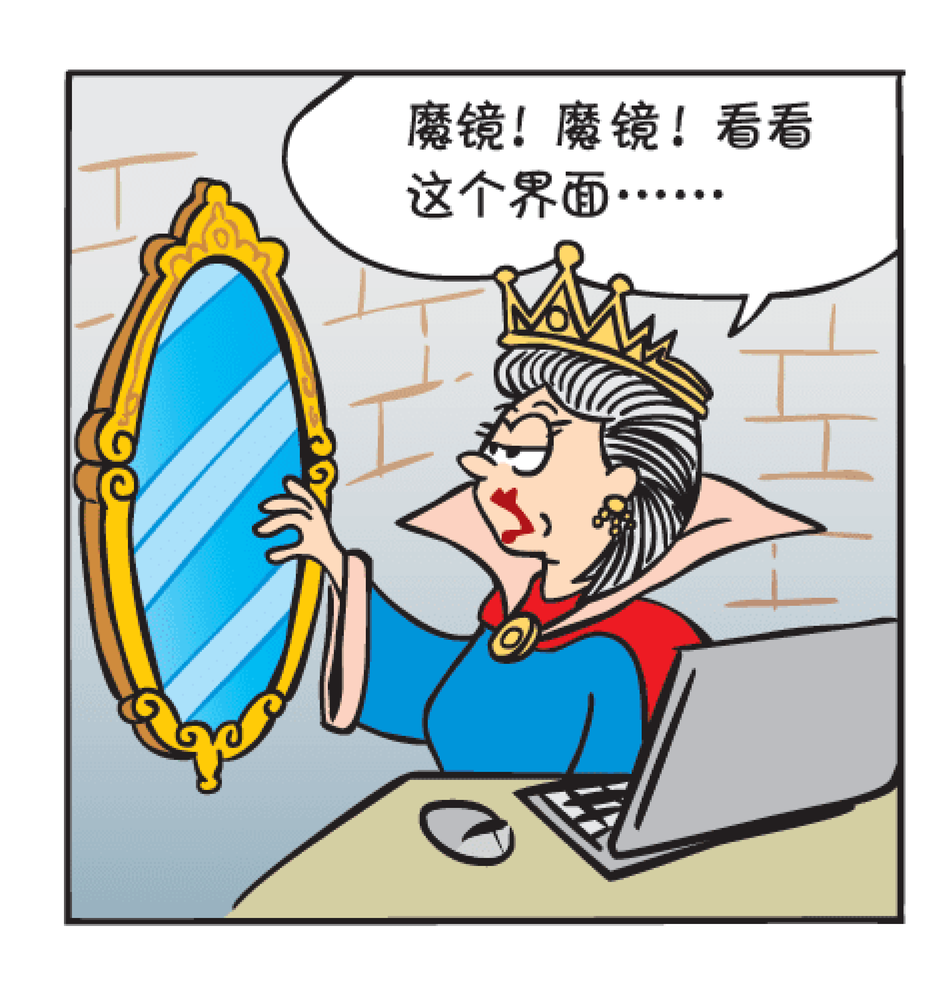
为了帮助用户学习使用产品，或者解决实际问题，你也应该提供易于检索、便于逐步学习的帮助信息。在必要的时候还可以提供交互式的逐步学习资料，甚至是实际练习。

尽管我之前提到可能很少人使用帮助文档，但是你还是不知道到底会有多少人想用。而且，帮助文档是一个有效的保护机制，一个以防万一的措施。它有时候甚至还能让你直起腰板面对那些脾气不好的投诉用户：“对不起，但其实我们在帮助里面已经说得很明白了。”

1.3.5 如何粗略地评估可用性

当你辛辛苦苦设计完一些方案后，有时候可能心里没底。这种时候先自我验证一下设计可用性的好坏，能够事先发现一些问题并解决，从而避免在以后真正的测试和评估中漏洞百出，招人笑话。

那么，如何来自己评估？



先自我评估一下可用性

这就涉及到刚才提到的可用性和用户体验的目标，以及把可用性目标概念化后得到的可用性原则。

首先当然是产品的实际功能是否能满足用户的需求。这是最基本的前提，也就是可用性目标中的“可行性”。然后根据产品的特性和针对用户群体，指定它的用户体验目标，把这些用户体验目标和所有的可用性目标相综合，再列举出来。

我们可以把这一过程分为5个步骤：



根据第一印象总结该产品工作方式的长处和短处，并且列出它的功能。列出一般用户将用它来执行的任务。让我们来看看，这款产品的实际功能与用户的期望相比，是更多、相同还是更少？



列出评估该产品的可用性目标和用户体验目标。根据目标的重要性进行排序。（当然你要说出这么排序的道理）



把选择出来的核心可用性目标和用户体验目标表达成若干个问题的形式，再根据这些问题进行大致的评估。“软件采取了什么具体措施来保证安全性？”、“网站的一些功能对于普通用户来说是否比较容易掌握？”这些是关于可用性目标的问题；而“使用它是否让人感觉有趣？”、“用户会不会在使用中常常觉得沮丧？”这些则属于用户体验目标的问题。



根据上一节里面提到的那些可用性原则，看看这个设计是否能够满足那些原则的要求。



根据所做的可用性评估，提出界面的改进方案。

当然，面对这么多的设计原则，我想你很可能会面临一个权衡的问题，因为你很难设计出一个能适应所有原则的十全十美的界面。例如为了让页面更具启发性，有人可能会故意把外观设计得类似真实世界（比如可能会把植物学网站做成原始森林那样子）。但这样就会使用大量的图片，导致页面下载所需的时间延长。这需要你经常全面性地考虑，然后进行取舍。

“一致性”是另一个可能造成问题的设计原则。比如说，有人会把网页里所有的超链接都设置成一个颜色。这样确实非常有一致性，但在美观方面就会大打折扣。而且有时候 A 的一致性还可能会破坏 B 的一致性。

甚至在某些时候，不一致的界面反而要比一致的界面要更容易使用。

我们举个不是交互设计的例子来说明这个情况。一个家庭往往有很多书，最简单的方法就是把它们集中放在书柜里面。这样，你只需要使用这个简单而一致的规则，就能够很方便地找到它们。这非常“一致性”。

但是，对于那些无法放进书柜，比如很大的画册，该怎么办？你可以把这种画册单独放在某个柜子里或者陈列起来。日常的期刊和杂志？你可以放在茶几下面，这样没事了就能翻翻。当然还有一些其他的书，你可以把计算机工具书放在书桌的电脑旁边，把睡前小说放在枕头下面。显然，先前的一致性规则已经不再适用。



你会这样在家里放书么？

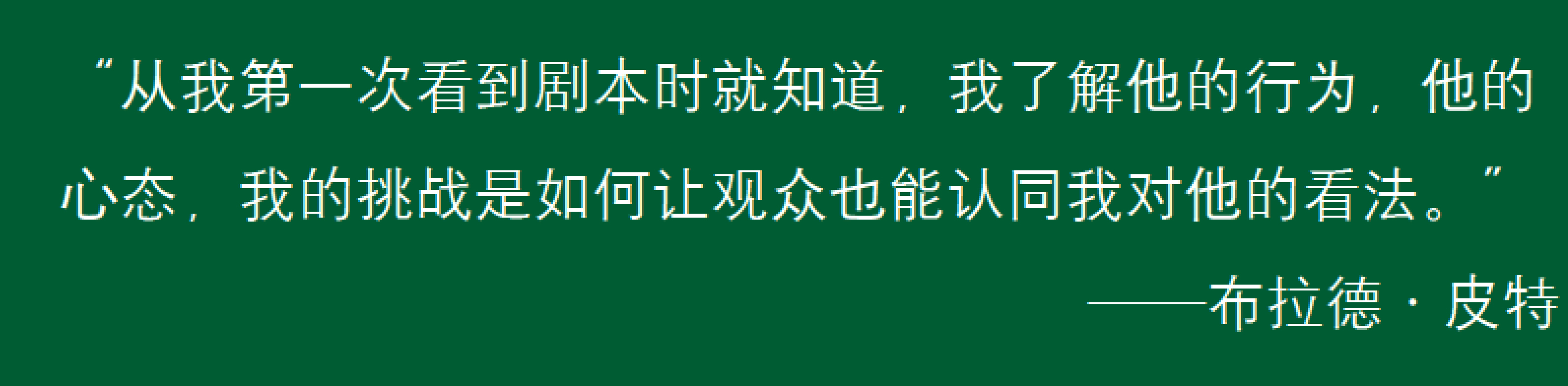
其实，增加放书的地方，实际上就是引入不一致性。这样确实会增加学习时间，也就是“学习”各种书都放在何处的时间。但是，把书放在不同的地方，通常会使得它们更容易找到，因为它们就在使用场所的附近，而且靠近它们作用的对象（例如软件工具书就放在电脑旁边）。

同样，设计界面也是如此，引入不一致性可能会导致界面更难学，但有时候它反而会使得界面更易于使用。所以你应该全面考虑所有问题，并且结合实践经验，对各种设计原则加以权衡和取舍。



第二章

了解用户，了解需求



“从我第一次看到剧本时就知道，我了解他的行为，他的心态，我的挑战是如何让观众也能认同我对他的看法。”

——布拉德 · 皮特

交互设计是要**优化**人们与计算机产品的交互。

怎么优化？这个问题暂时先放在一边。让我们先想想优化给谁。

当然了，这是傻子都知道的答案："谁用这个产品，当然就优化给谁"。

没错，我们当然是优化给用户。但是如果你不知道用户是谁，也不知道他们想达到什么目的，那你怎么知道该优化什么，又该如何优化呢？

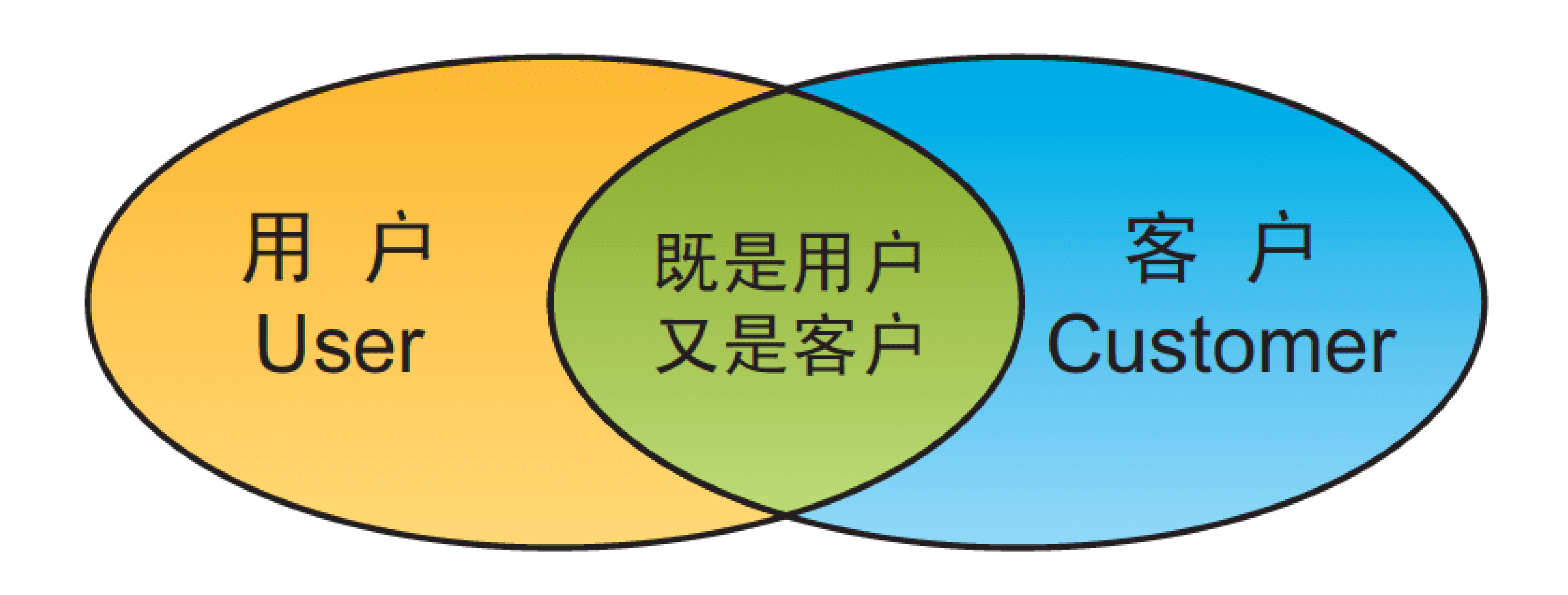
所以，一切的出发点应该是：找出谁才是你需要了解的人，然后弄清楚他们想要什么。

前面说过，"理解用户的需要，从而建立明确的需求"是交互设计的基本活动。我们就是要将用户时时刻刻摆在所有过程的首位。这就是"以用户为中心"的方法。

2.1 谁是用户？

找出用户似乎是一件很容易的事情。但首先我们需要弄清楚"用户(user)"和"客户(customer)"的概念。没错，产品的销售对象是"客户"，只有客户才是愿意掏腰包购买产品的人，考虑他们的感受是非常重要的。但是客户并不就一定会亲自使用我们设计的产品。

换句话说，客户不一定就是用户。



举个简单的例子，某个网站制作公司的老板不一定必须得是网页设计师。他确实需要购买一些软件产品用于开发网站，比如 Dreamweaver、Visual Studio，甚至包括 Flash 和 Photoshop。他购买了这些软件，他便是 Adobe 或者微软的客户。但老板并不一定要去使用这些软件，甚至可能根本就不懂如何操作。也就是说，他没有与这些产品发生直接的交互，自然不是软件的用户。软件是否容易使用，界面是否友好，有没有良好的通用性，这些跟他没有多大关系。

但是，他手下的那些网页设计师每天都面对着这些软件，他们就是软件的用户。而交互设计的目标正是为了支持这些人（他们很辛苦）的需要，满足这些人的期望，让他们在工作中能够更方便、更舒心和更有效。

当然客户也可能就是用户，例如我们绝大部分人都使用微软的 Windows 操作系统，我们就是它的直接用户；而正因为我们购买了它，就能以客户的身份要求微软进行服务。

所以，我们说交互设计的核心是“以用户为中心”，而不是“以客户为中心”。也许对市场和销售部门来说他们认为客户就是上帝，但对于用户界面设计，上帝不是别人，而是“直接与产品交互、完成某个任务或达到某个目的”的那些人。

所以，在你确定谁才是产品的目标用户时，必须对市场和销售部门的想法加以过滤或者补充，从而得出对你自己最有用的信息。

2.1.1 普遍而又特殊的用户

我们需要注意的是，“用户”这个概念既有普遍性，又有特殊性。

用户的普遍性

我们的用户当然指的是人类。那么他们也应具有人类的共同特性，在使用任何产品的时候都会在各个方面反映出这些特性。人的行为会受到他们本身基本能力的影响，例如视觉和听觉、记忆力、分析和解决问题的能力，还有对刺激的反应能力等等。与此同时，人的行为还会受到心理和性格取向、教育程度、以往的经历，甚至所处的地域环境和文化背景等因素的制约。

用户的特殊性

与此同时，交互设计和评估所研究的“用户”又是一个特殊的群体：正是**这些人**与我们的产品使用相关，而不是其他人。他们可能正在使用我们的产品，也可能是潜在的用户、会在将来某个时候使用我们的产品。

这些人在使用产品的过程中，会表现出一些与产品有关的特征。例如对于产品了解多少、对产品的期待是怎样的、是否拥有使用产品所需的基本技能，以及他们是经常使用该产品还是偶尔为之等等。

两者兼顾

所以，我们在了解用户的过程中，既要顾及他们**作为人类的一般属性**，又要考虑他们**与产品相关的特殊属性**。这两方面缺一不可。

比如说，现在我们要设计一个女性购物的网站，我们就需要知道用户群体的普遍属性，包括：

- 大部分用户为女性，女性喜好较为温馨、可爱和时尚兼顾的视觉效果；
- 一般而言女性分析和解决问题的能力较弱（例如推理和归纳），因此设计中不要太强调逻辑思维，尽量偏向于感性；

- 在知觉方面，女性高于男性，她们阅读、领会的能力较快。但在对细节的知觉方面不如男性准确。
- ……

而用户的特殊属性方面，即他们与本产品相关的方面，需要注意的比如有以下几点：

- 用户光顾网站的目的是为了赠送礼品还是自己购物？
- 用户是希望购物更方便、节省时间，还是希望能更详细地对比价格？
- 用户是否经常光顾类似的购物网站？他们对此有没有丰富的经验？
- ……

2.1.2　用户可不是越多越好

我们必须坚决抵制这种诱惑，即总想把目标用户的基础定为“每个人”。没错，我们都希望产品能够大卖，获得更多的经济利益。所有公司都想要拥有最广泛的潜在市场。这完全无可厚非。

但是，试图让“每个人”都竖大拇指的产品，很可能到最后没有一个人竖大拇指。有句古话叫做“贪多嚼不烂”，多而不精是我们必须极力避免的。什么都顾及的结果必然会导致各个方面都需要相互做出让步，这样很可能每个人都不会感到十分满意。



所以，找到目标、竖立靶子才是最开始需要做的事情，也就是要清楚地认定谁才是目标用户。

这其实并不难。因为某个产品的用户，通常都具有某些共同特征。也就是说，你可以选择一个特定的人群作为目标用户的基础，然后把注意力放在他们身上。

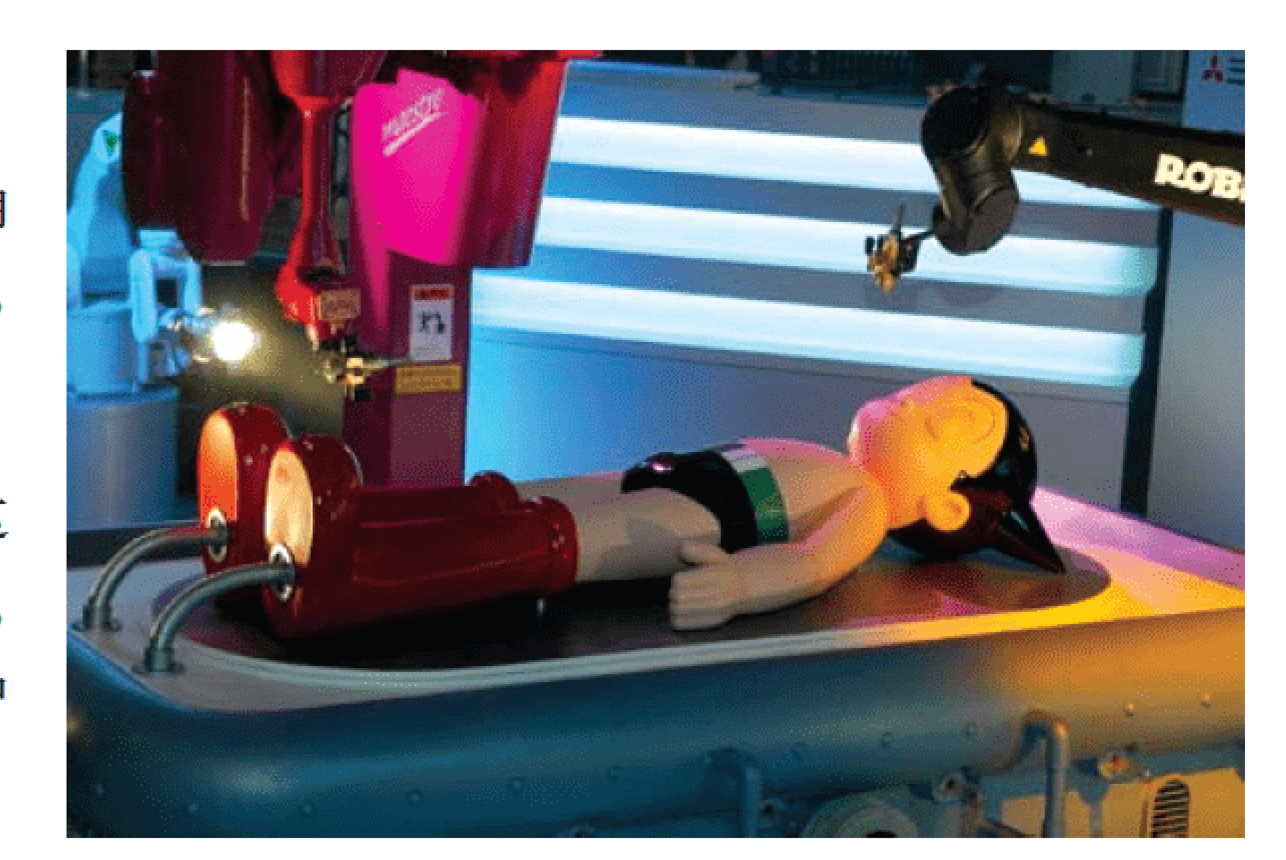
当然，也许这个产品还有其他类型的用户，有时候你可能得做出艰难的取舍。例如用户很明显分为两种。在这种情况下，要么你就分头考虑（但是效果不一定好，除非你分别为这两类用户都开发一套版本），要么就根本不考虑其中一种。关于这一点，你也可以问一下市场和销售人员的想法，然后再判断。因为这些往往都涉及到公司的战略性目标。

2.1.3　“阿童木”的诞生

为了准确地定义用户，有时候我们会根据调查建立一套标准的“用户模型”。这其实是一种非常好的方法。所谓“用户模型”，就是描述用户的一个模板，通过这个模板你就能够把用户的共性给总结并描述出来。

为了便于理解，你甚至可以根据这些特征“虚拟”出一个标准的“角色”出来，这个“角色”就代表着用户，或者某一类用户。我们应该把他当作是我们的一位朋友，真正地生活在我们周围，因为我们知道他长得什么样，喜欢什么、讨厌什么，每天都做些什么事情。融入他的生活，你就能了解他会有什么样的需求，就能发现什么样的产品才是他最需要的。

这就好像我们在建造“阿童木”一样。



为了了解用户，我们可以根据这个“角色模型”来寻找特定的用户人群。简单来说，就是寻找跟这个虚拟人相类似的群体。

这需要我们分析用户的各种特征。在这里我给出一些常用的描述用户特征的方面，你可以根据它们建立一个比较完整的“用户模型”。

- **一般数据**

这些包括用户的年龄、性别、教育程度以及职业特征等等。

- **性格取向**

例如用户的性格是内向型还是外向型？
还有，他是形象思维型还是逻辑思维型？

- **一般能力**

用户的视力、听力等感知能力如何？有没有色盲或者色弱？有没有可能他左耳的听力比右耳要差？
他们的判断和分析推理能力如何？能否举一反三？
他们的体能如何？习惯用右手还是左撇子？

- **文化区别**

用户们主要生活在哪个国家或地区？他们的宗教信仰是什么？
他们是否把英语作为第二语言？他们写字是从左到右还是从右到左？
他们的民族习惯和生活习惯如何？他们喜好或者反感什么？
他们和他们的父母或者孩子之间有没有代沟？

- **对产品相关知识的现有了解程度和经验**

用户能够轻松自如地阅读屏幕吗？老年人经常会觉得计算机屏幕很晃眼。

他们会不会使用键盘或鼠标？
他们使用过类似功能的系统没有？
他们有没有与系统功能相关的知识？例如某个记账的软件可能需要一点财会的经验。

- **产品使用**

用户只在办公桌旁边使用产品吗？或者他们回家后躺在床上也会使用？
他们使用产品有没有规律？大概的使用时间和频率又是什么？

- **环境和技术基础**

用户公司的网络速度如何？是宽带还是 Modem ？
他们一般使用的显示器的分辨率是 800×600 还是 1280×800 ？显示器不会还是 256 色的吧？
他们是使用 Windows 操作系统还是 Mac OS? Win 98 还是 Vista ？
他们的电脑有没有外接音箱？有没有手写板？是无线笔记本上网还是几个人抢一条网线？

通过上面这些用户特征，你就能建造自己的标准“阿童木”。

显然，对具体的某个产品来说，定义用户并不一定要对他们所有的特征进行描述。但是逐一审视这些特征，将会帮助你更为全面地把握设计的可用性，从而不会遗漏。

设计带有显示屏的公用电话时，我们可以定义目标用户为所有大众。但与此同时，了解到用户所在的地域或国家就意味着需要考虑一些显示界面的问题。

例如在阿尔及利亚、埃及、埃塞俄比亚等国家主要使用的是阿拉伯语。阿拉伯语是从右到左书写的，那么出口到这些国家的公用电话显示界面就必须适应当地人的使用习惯。而了解用户以前是否使用过这种比较复杂的多媒体终端，就直接影响到产品的界面设计、工业造型设计，甚至是用户手册的用语风格。

前面说过，有时候用户可能会分为好几个类别，尤其是那种面向范围非常广的产品，比如说门户网站和在线游戏平台。如果你希望每个类型的用户都抓住的话，就需要分别为他们建立用户模型，用以描述每个类别的用户。

有时候我们可以根据年龄层次可以把用户划分为青年、中年和老年用户。这些用户类别之间的差异比较大，所关注的对象也不一样。例如在媒体网站中，青年更喜好娱乐和高科技方面的资讯，而中老年用户则偏向于政治、财经等方面。而在线休闲游戏网站里，青年用户更喜欢刺激、活泼的对战游戏，中老年则偏好桥牌、象棋或围棋等类型。

除了年龄差异之外，你还可以根据性别、收入或知识水平等方面对用户进行划分。至于用什么划分标准最为合适，你可以结合市场部门对用户群体的区分来加以参照。

2.2 什么是需求？

哪怕已经跨入了 21 世纪，如果在头几年我们询问人们在网络中“需要”什么时，可能没有很多人会提到 Google 这种方便的互联网信息搜索工具，或者 YouTube、土豆网这种全民级视频短片分享网站。但如果我们先提出这种可能性，再问他们是否愿意使用时，回答可能就不同了。

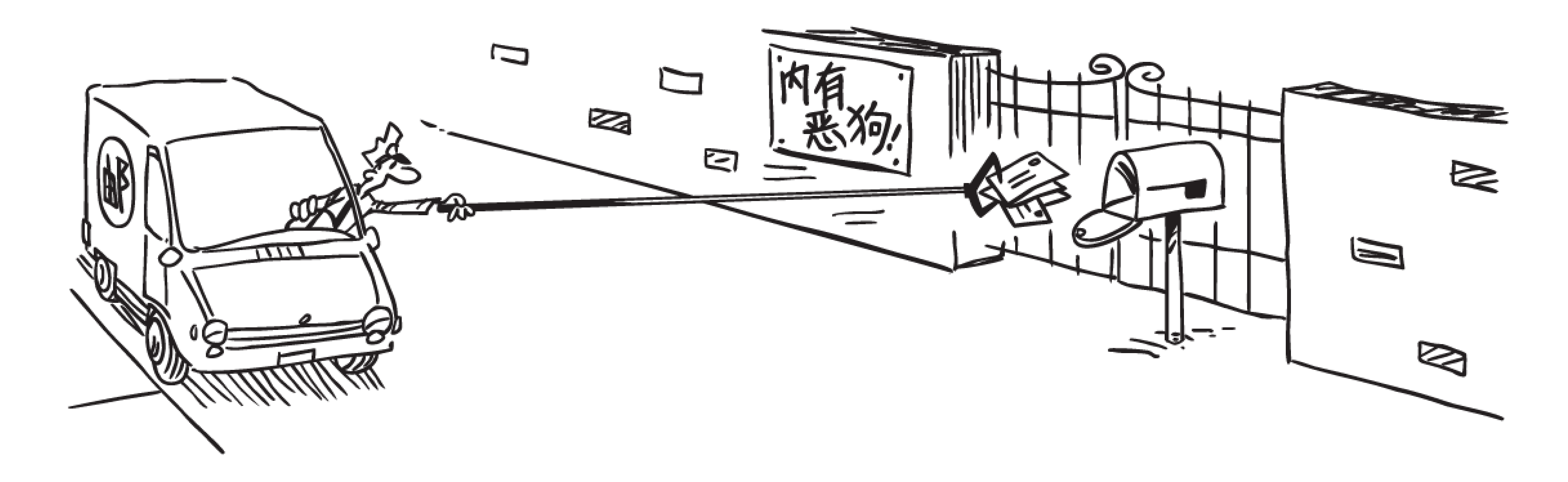
可见，为了找出用户的需要，我们不能简单的问：“你需要什么？”然后提供它，因为人们未必知道什么是可能的，有些需求甚至可能是用户“做梦都不曾想到”的。

要了解需求，更合适的方法是：我们必须理解用户的特征和能力，了解他们想要达到什么目标，准备如何达到这些目标。另外还需要知道的是，如果我们提供另一种支持，他们达到目标的效率是否会更高。

2.2.1 “需要”产生需求

在许多方面，每个人的特征和能力都有所不同，这就导致用户的需求很可能彼此不同。

很简单的例子，人的身体特征就会影响设计：针对色盲用户的产品就必须考虑色彩的问题，视力不好的用户也许希望能够调整系统字体大小。因为环境给他们造成了影响，所以他们希望克服这些不便。就像下面这个邮递员一样。



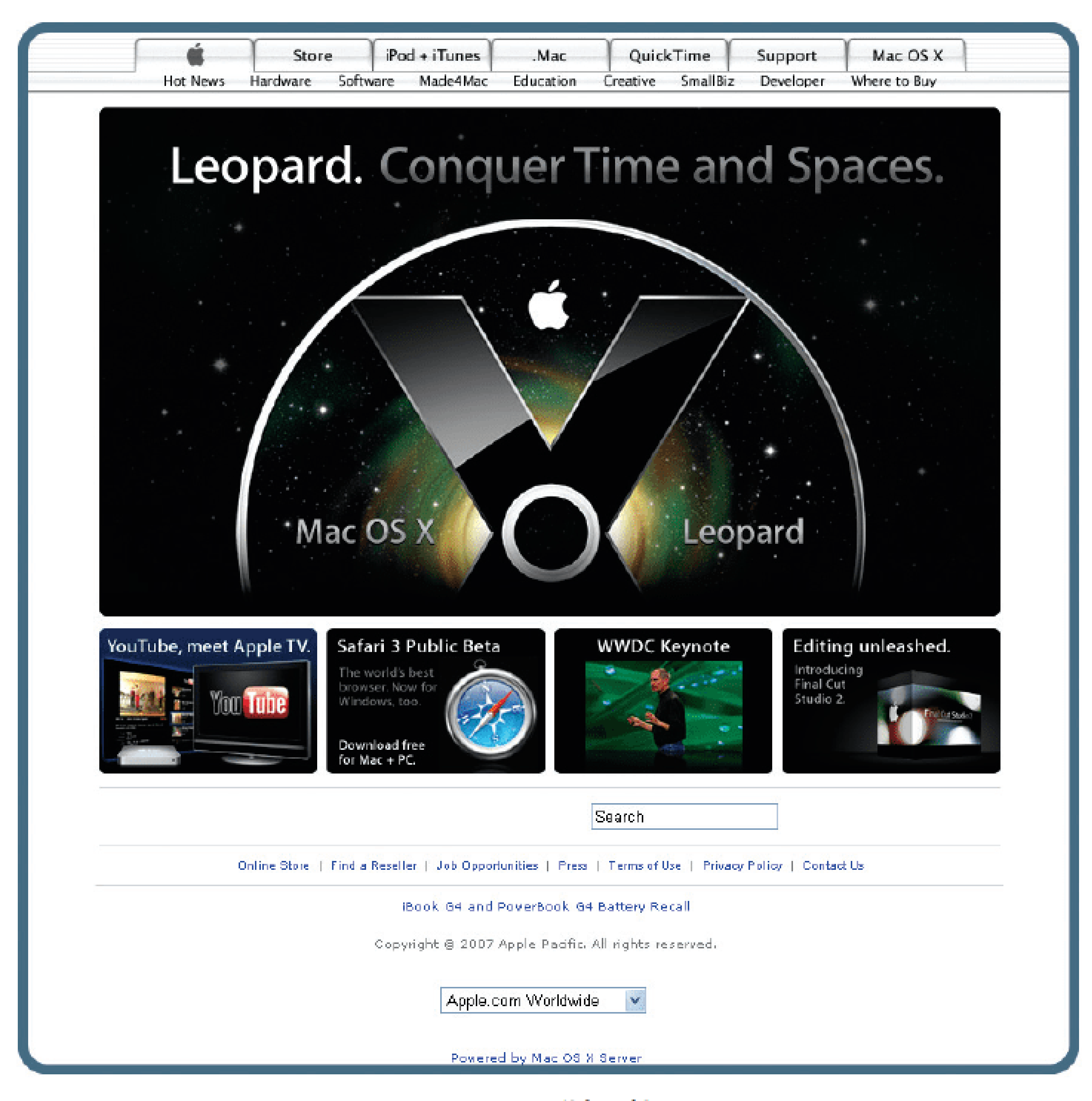
另外，文化和经验的差异也会产生需求的差异，比如可能习惯使用不同的术语，或者对技术的接受程度不相同等等。

很多国际公司的网站都有不同国家和地区的版本，除了语言问题之外，很重要的一点就是为了适应不同地域用户的不同需求。

比如下图中的 Apple 美国网站和 Apple 亚洲网站。我截屏的时间仅仅相差几秒钟，但我们可以明显看到其推出的产品、服务和功能的不同。



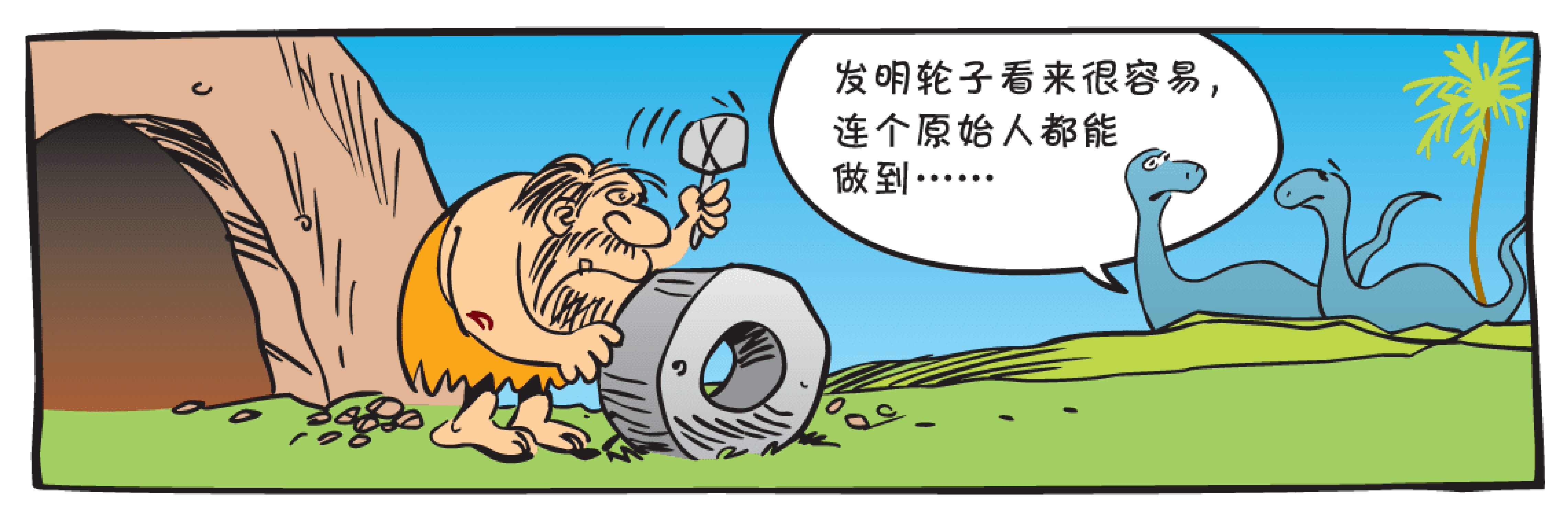
Apple美国网站



Apple亚洲网站

另一方面，我们也可能需要设计出具有开拓性的产品，从而满足那些甚至从没人预见过的需求。这些也许是划时代的、前所未有的产品。如果开发的是这样的创新产品，确定用户和有代表性的任务就比较困难。

在发明微波炉之前，你根本找不到用户询问需求，也不存在什么有代表性的具体任务，比如用它来爆玉米花。所以发明微波炉的人就必须自己去想象将有谁会使用它、用它来做什么。



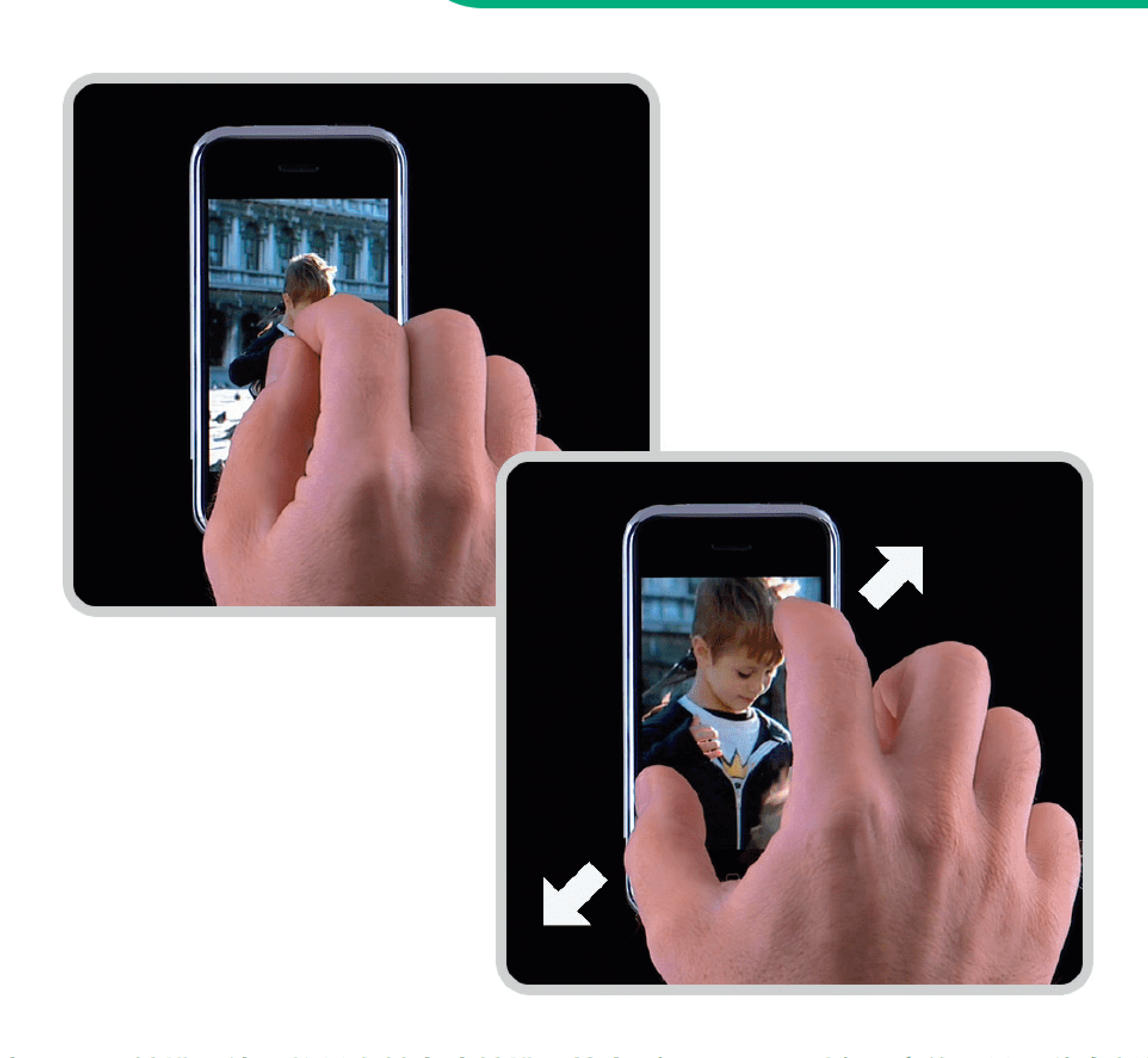
发明轮子也是一样

在这种情况下，当前或过去的行为是对未来行为的一种很好的启示。我们可以从理解相似的、已经存在的行为入手。

在发明手机的时候，并没有特定的用户，也没有什么具代表性的任务，只有标准的普通电话。而研究普通电话就是一个很好的开始，有助于理解人们使用它主要执行什么任务，以及有哪些相关任务。除了能随时随地打电话之外，用户还需要查找电话号码、发送和保存留言，还有找出刚打来电话的人的号码等等。根据这些行为，手机中又加入了电话号码存储器、手机短信息服务等功能。

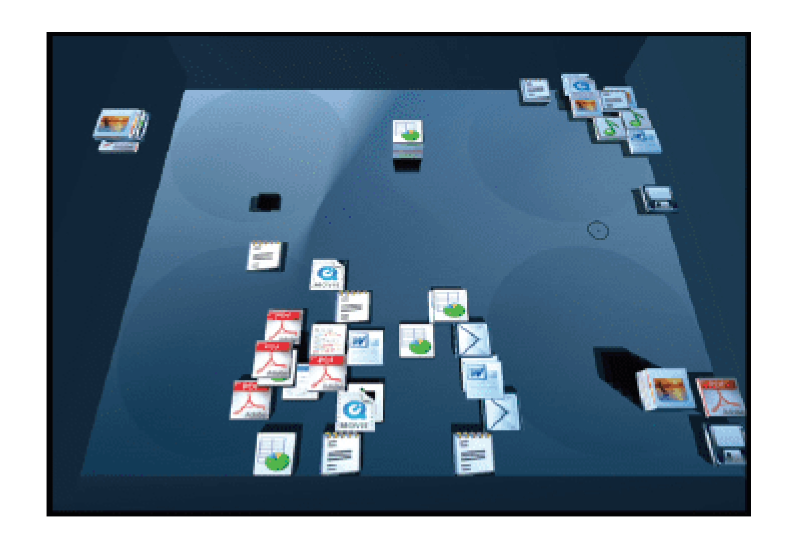
Apple 公司的最新产品 iPhone，在某些方面就很具有开拓性。这是一款带触摸屏的手机。一般这类手机都需要用触感笔操作，但是 iPhone 为了更好地支持用户，强调只用手指就能完美地控制。这样用户在任何时候从口袋里拿出 iPhone 就能使用，不必再费事地抽出触感笔，也不必担心触感笔是否会遗失。只要手指还在，他就能使用 iPhone。

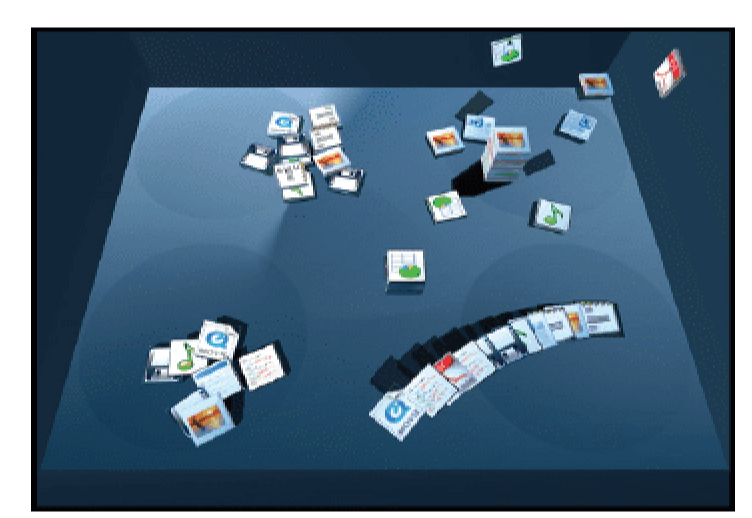
由于手机体积不能太大，因此在手机上浏览图片就会显得有些不方便。普通的触摸屏手机都需要点击图片工具栏的“+”或者“-”来缩放图片大小。似乎除此之外也没有别的办法。但是这个细节却被 Apple 公司的设计师们抓住了，他们进行了创新，给用户带来了从没有想到过，但是却十分新颖精彩的体验：用两根手指将图片“扯”大！仿佛用户拥有魔法，轻松地做一个手势就能达成目的。与此同时，用户在使用 iPhone 浏览任何网页的时候，也能够如此随意地放大页面。我相信很多用户在购买了iPhone 之后都会不停地尝试这个功能并且乐此不疲——因为确实很引人入胜。



其实 Apple 触摸屏使用的是多触点式触摸屏技术（Multi-Touch）。这并不是一种全新的技术，早在 20 多年前鼠标刚刚问世不久就有人提出了。也许即使 Apple 把这个地方做成普通的放大和缩小按钮，也不会有任何的用户会产生异议，但是也正是这些小细节的创新，使得 iPhone 在性能、造型等方面之外的很多地方显得与众不同，让用户使用和操作起来更为方便和愉悦，从而能够展示其自身独特的魅力，卓尔不群。这一点从 iPhone 发售日前世界各地 Apple 零售店门口的排队长龙就能得到证明。

除此之外，新技术一直在不停地涌现，世界上有无数希望有所作为的设计师和技术人员在孜孜不倦地尝试或开发着新的交互方式，有一些甚至是革命性的。例如 2007 年刚刚在蒙特利尔 TED 大会上演示过的 BumpTop。这是一个真三维、高度自由化的桌面系统。









当然这个系统还只是个原型。在这个系统中，你可以就像在真正的桌面上一样堆放文件，方便自如地翻找、归纳它们。虽说目前还很难想象这个技术有什么实际意义，但是它令人印象深刻，几乎 100% 地模拟了真实的物理桌面和书本，并且提供了很多方便的操作和指令，例如排序、查找、标识重要性等等。交互方式也非常有意思。

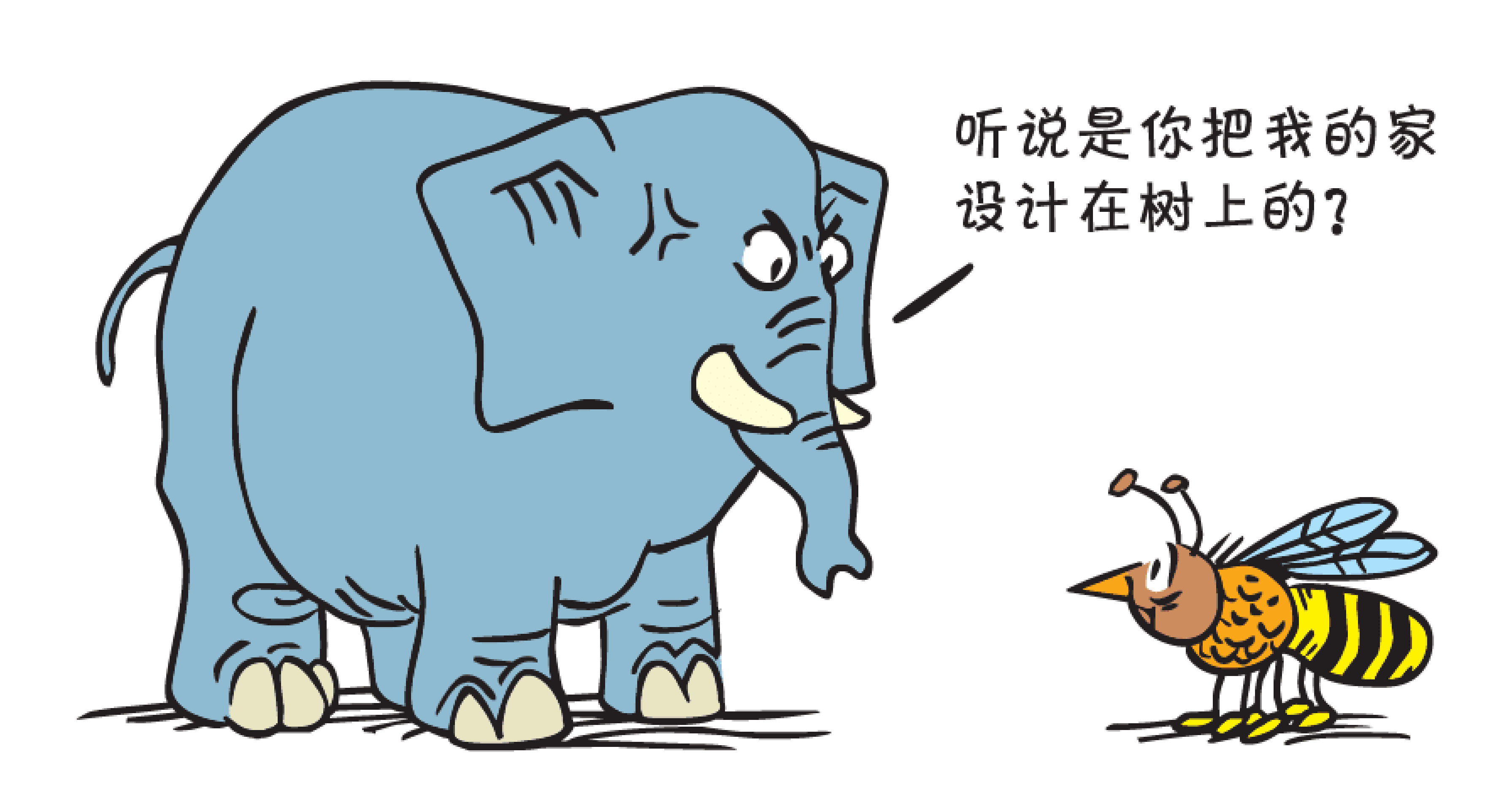
让我们向这些有才华的设计师们表示敬意。

2.2.2　点了牛扒，端上来的却是意大利面

我相信任何设计人员都会尽心尽力地设计产品。但是，有些时候他们可能会根据自己的意愿进行设计："我想做好，那么就按照我认为好的方式去做"。

然而，他们的想法与用户的想法未必一致。设计人员往往倾向于创建自己想要的产品，他们也不了解用户还想要些什么。因为是自己设计的，所以他们知道网站如何运行、什么时候在什么地方会显示什么信息、屏幕上的每件东西是什么意思。他们假定用户能够以他们所希望的感知和理解方式感知并理解一切。

但实际上，设计人员的专业背景决定了他们不可能充分理解普通用户的需求。可能对他们而言非常简单、无须思索的东西，却有很多普通人不理解。用户和设计人员之间的差异就好像下面这两只动物一样。



当人们第一次开始使用网站或其他什么产品的时候，他们几乎不知道它如何运行，也不知道屏幕上的一切都是什么意思。他们并不知道设计人员的个人意图。所以，开发者只关注自己的需求肯定过于狭隘了，他们必须咨询目标用户。

在项目开始时，可能已经有了一些初始需求。也可能没有，一切从零开始。不管初始情形如何，我们都必须讨论、提炼和澄清用户的需要、需求、期望，这包括了解用户及其能力、当前任务和目标、产品的使用条件以及对产品性能的要求。

如同“理解用户需要，建立明确需求”的字面意思所指出的，这一活动有两个目标：



尽可能地理解用户、用户的工作和环境，从而保证产品能够使用户达到他们的目标。



从用户的需要中提炼出一组稳定的需求，作为后续设计的坚实基础。

“理解用户需要，建立明确需求”这一活动本身就是一个迭代过程。它的各项子活动是相互影响的，并不是在持续几周或几个月后就完全结束。从用户那里得到了一些信息后，你可能又会发现需要更多信息，以检验或肯定某些想法。另外在实践中，用户可能也会随时发现新的可能性和新的需求。

记住，理解产品要做什么并且确保它能支持用户的需要，是至关重要的。如果把需求弄错了，你的产品要么就被用户忽略（这是最好的情形），要么就被用户鄙视（这是最坏的情形）。不管怎么说，这种结果对于你和购买产品的客户来说都是不利的，就好像你点了牛扒，我却端上来一盘意大利面。

所以，我们必须坚持“以用户为中心”。如果正确听取并考虑用户的意见和需要，产品就更有可能满足他们的需要和期望。

有些需求是已经存在的，你只需要列出来就行。比方说，客户需要制作一个女性购物网站，那么很明显在一开始就有些需求确立了：基于互联网技术、需要与后台数据库衔接，诸如此类。

而另外一些需求可能用户知道，但是需要你去询问。例如客户关注的到底是网页下载速度（图片量就不能太大）还是视觉效果（图片量就不能太小）？用户们对这种网站有没有使用经验？等等。

还有一些需求用户可能知道，但表达得并不十分明确。或者即使他们提出了需求，也没有仔细地研究过。这种情况就需要你进一步去分析。例如客户要求“女孩们应觉得这个网站吸引人”，这确实是一种需求，但有些不精确。你还需要进一步调查女孩们到底喜欢什么样的网站。再比如客户希望“网页下载速度快”。那么到底多快才算快？如果进一步把它定义成“完整下载网页的时间不应超过 5 秒”这样的具体标准，才算是一个明确的目标。

所以说，在收集需求的过程中需要强调“可衡量性”。也就是说，对于每一个需求你都应该画个问号：我怎样能判断这个需求得到了满足？

判断满足与否，也分为几个方面。

- 有些需求可以用很客观的方法进行衡量。例如产品是否具有某种功能，或者网页下载的时间是不是在限定范围之内。
- 另外一些需求则可以通过用户测试等方法来衡量，例如用户的满意程度。



- 还可能有一些需求由于种种原因无法被直接衡量，比如对效率的提高，或者对用户精神的鼓励程度等等。你也可以把它们转化为可以直接衡量的内容。比如说，如果以前 2 个月才能完成的项目，现在一个月就能搞定，那么就表明效率提高了。
- 至于那些完全不能衡量的内容，删除掉，不要考虑它们。因为你确实没办法知道这方面到底做好了没有，那么就不要耽误时间。

2.2.3 各种类型的需求

需求也分为不同的类型，这样我们可以把它们分类归纳，看起来也清楚一些。

在这里我列出了一些不同的需求类型。这并不是非常严密的分类，但起码能够提醒你至少需要了解哪些方面。

功能需求

也就是说要了解系统应做些什么。例如“系统应支持及显示多国语言文字”就是一项功能需求。它也可以分解得更为具体，详细指定要求哪些国家的文字，如英文、韩文、德文等。甚至还可以再具体一些，就像“应提供英、韩、德三个国家的文字，每种文字至少三种字体，而且都带有粗体、斜体和标准体”。

用户需求

这要求我们了解目标用户的特性，尤其要考虑他们的能力和技能。比如说，新用户可能需要循序渐进的提示，所以界面应该更醒目、带有明确的信息；而专家级用户则希望更灵活的交互方式（例如快捷键）和更多的控制权。

数据需求

这些是关于数据的类型、大小、持久性、和准确性等方面的需求。绝大部分交互产品都需要处理或多或少的数据。例如一些在线购物网站，或者银行账户管理系统，就一定要保证数据必须准确，而且保密性高。

环境需求

这里指的是产品的使用环境。对于 Web 和 Web-based 产品来说，我们主要关心的是技术环境。比如说，这个网站在 Apple Mac OS 操作系统里面是否也能完美展示？不同的浏览器呢（IE、Netscape、Apple 的 Safari、火狐、傲游或者腾讯的 TT）？用户的网络条件如何？开发和使用中又存在着哪些技术限制？








可用性需求

我们还需要了解产品的可用性目标和评测的标准。在开发初期，我们就应该确定如何去评估可用性，而且要协商一致，之后用它来衡量产品在这方面的质量。

这些需求经常会相互冲突。

Apple 的 iPhone 提供了许多功能，包括打电话、收发短信息和电子邮件，以及浏览网页、听音乐、看电影等等。但同时它又是一款便携式的手机，只能使用 3.5 英寸的小屏幕，而且没有键盘。这就是它的限制。限制意味着会出现冲突。

这些冲突包括如何在480×320像素的屏幕中浏览正常网页的问题。为了避免给用户带来不便，设计人员就必须仔细考虑什么样的解决方案是最合适的。他们的解决办法是：用户正常浏览时显示的是缩小的网页，从而可以大致了解网页概况，也便于上下拖动。而同时他们也可以通过某种操作（用手指双击屏幕）对网页进行放大，用以阅读文字、查看图片等细节。另外，当用户横向放置iPhone时，网页也能自动放大显示。



另外一个明显的冲突是如何用手指输入文字。也就是当点击字母按键的时候，可能会因为手指挡住了部分按键而导致输入错误。设计人员也想出了一些针对性的策略：1. 点击字母后会出现放大显示，能保证你知道自己点击了哪个字母；2. 提供智能纠错功能，哪怕你输错了字母，系统也能正确显示你所希望输入的单词。



所谓“设计”，就是提出各种方案，不断地做出选择和决定。所以你不得不想办法去减少这些冲突带来的影响。

2.3　如何确定需求

我想你已经认识到，“了解需求”不能闭门造车，需求并不是设计人员自己想象出来的。我们只能去与用户交流，使用科学的方式去搜集充分的相关信息，从而找到并且提出一组稳定的需求。

你也许会说，我已经知道了需要哪些需求，那么如何得到它们呢？把所有用户都找出来一个一个地去问吗？

那肯定是难以想象的。事实上也没有这么笨的办法。

需求信息的实际搜集方式是多样化的，而且非常灵活，各种技术也可以相互组合和扩充。这些技术并不难，主要包括问卷调查、用户访谈、观察用户以及集体讨论。它们都需要用户们的积极参与，所以这也体现了“以用户为中心”的核心理念。

在搜集需求信息的过程中，你还可以使用各种“道具”，例如任务描述和产品原型，它们可以让这个阶段更加简单。在后面我将告诉你如何制作这些“道具”。

2.3.1　介绍一些数据搜集的方法

这里所说的数据搜集技术，其实就是从用户那里获取需求信息的技术。

这些技术非常重要。你在现阶段确定需求时需要它们，日后在可用性测试和评估中也会用到。也就是说，这些技术贯穿着我们的整个设计。我一再强调交互设计的核心观点是“以用户为中心”，就是时刻需要考虑用户的特征和需求。只要你想从用户那里得到有用的信息和数据，你就只能通过这些数据搜集技术。

不过不要被“数据搜集”这个貌似很专业的词汇吓倒。其实它们都是些很常见的方法，我们每个人都可能经历过其中的一些。比如说 ：

问卷调查

这个相信你很熟悉。所谓问卷调查，也就是设计一系列问题，通过网络（如电子邮件或网站）、电话或者纸张（包括面对面及邮寄）形式进行大面积铺放，然后把问卷回收再导出特定的信息。如果是纸张形式，你也可以在展览会、餐厅、工作或者会议等场所发放。



主妇网的电话问卷？没问题，但请等我先把活干完。

对于 Web 和 Web-based 产品来说，通过网络当然是最方便直接的办法。

问卷的问题可以有多种形式，例如简单的选择题或是非题，你也可以要求答卷人对某些问题做出较长的说明，或者提供具体的意见。

在设计问题时，首先你要清楚这次调查的目的是什么，然后对每一个问题进行仔细的审核。目的是要保证它们的含义清晰并且容易回答。

在问卷设计完之后，最好先找一些人试答一下。这是因为一旦问卷大面积铺放之后，即使发现了问题也很难再更改了。

在设计问卷和分析结果时，可能任何一点细节都会影响到结论的准确性和可信度。比如下面几个方面：

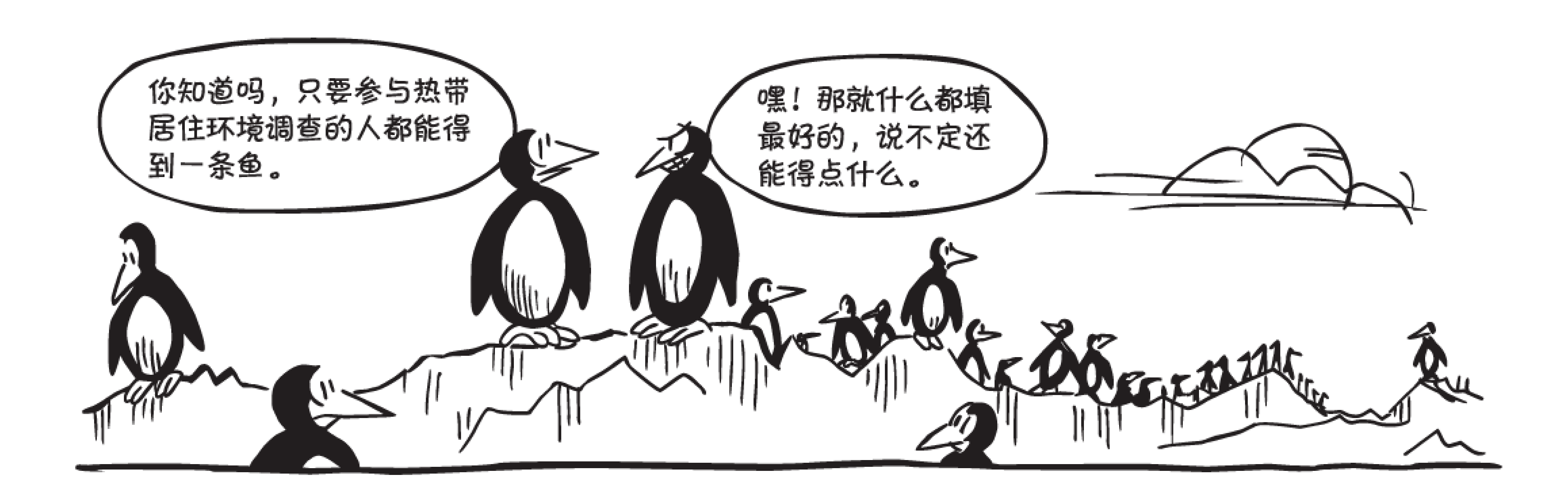
- **参加调查的用户不一定就代表所有用户**

  有时候当自愿参加问卷调查时，参加者可能有某类共同的心态，所以恰恰只代表了某一类用户，而不是用户全体。

- **用户参加问卷调查的动机可能会影响研究结果**

  有些人在参与有偿调查时会比较兴奋，或者会影响他们的客观判断。有些用户为了表示友

好、让调查方高兴而答题，甚至去猜测问卷人所期望的结果。这种问卷回收起来就没有很大意义。



- **问卷的来源可能会影响研究结果**

  也许有些人对某些组织或单位有特定的看法，用户在回答问卷的时候可能会受到这些观念的影响。如果你很讨厌某个公司，可能在填写他们问卷的时候心情会很不好。所以问卷中应当避免流露具体出处。

- **研究问题的措辞可能也会影响研究结果**

  也许用户在回答某个是非题的时候，并不“同意”，也不“反对”。他可能有些话想说，但你没有给他机会。所以问卷的选项应当平衡，也要注意在两个极端之间的情况。

问卷调查主要适用于调查人群分布很广的情况。因为你无法和他们一一面谈。但是这种方式没有足够的可信度和权威性，所以通常你需要和其他技术配合使用。比如说，用户访谈。

用户访谈

访谈其实就是个人采访。你和某人面对面的交谈，向他提出一组问题。当然你也可以通过电话访谈。

访谈最好是在用户的工作环境或者家里面。在这些地方人们比较自在，也能更从容地谈论自己的活动。比如平常都做些什么，使用什么系统或者设备。环境也能提醒他们某些事情，例如在接收邮件时遇到的问题之类。在其他场合他们有可能就会忘记这些事。

进行访谈之前，你和被访者都应该做一些准备。你需要准备一些问题，而且预测用户可能的反应。用户则可能需要将一些参考资料或实物带到访谈的现场。访谈时提出的问题可以很广泛，主要是挖掘他们对各种问题的看法。



—您觉得Google Dog确实很有必要吗?
—汪!

与单调的纸张或电子问卷调查相比，访谈更能激发被访者的响应，整个过程也更令人愉快。不过用户访谈非常耗时，而且要访问到所有想访问的人也不太现实。对这种情况来说，我们可以先访谈一些有代表性的用户，然后对更广的用户进行问卷调查，以验证通过访谈得到的信息。

观察和提问

人们有时候可能很难解释清楚自己在做什么，也很难准确描述如何执行某个任务。所以你可能需要花点时间来观察他们是如何执行任务的。你要了解他们在自然情形下的工作过程，同时做记录和提出适当的问题。这有助于你更透彻地理解用户和任务。

对观察用户来说，你可以自由掌控介入的程度。这个程度可以从根本不介入他们工作的局外观察，到与他们共同执行任务。

有几个方面你应当注意 ：

观察的人应当是最有代表性的用户。

要想更透彻地理解用户，你就需要在他们的工作环境下观察。

提问要具体，重点放在他们正在做的和刚刚完成的事情。

你在这一过程中产生的想法应该马上反馈给用户。这是为了验证你的理解是否正确。

有一个经常使用的方法叫做“有声思考”。“有声思考”的意思是，让用户在完成任务的过程中，

尽量清晰地口述当时思维的内容。这种口述进行得越及时，它的内容就越能准确反映任务的实际过程。

很多人不太习惯这种自言自语式的“有声思考”。他们会觉得很难为情，或者忘记需要说出来。在需要集中注意力完成任务的时候，边说边做也是很难的一件事情。在这种情况下你可以适当地提出一些问题，促使他们积极口述。提问时要多问中性、积极的问题，而且要保持灵活性。例如这些问题：



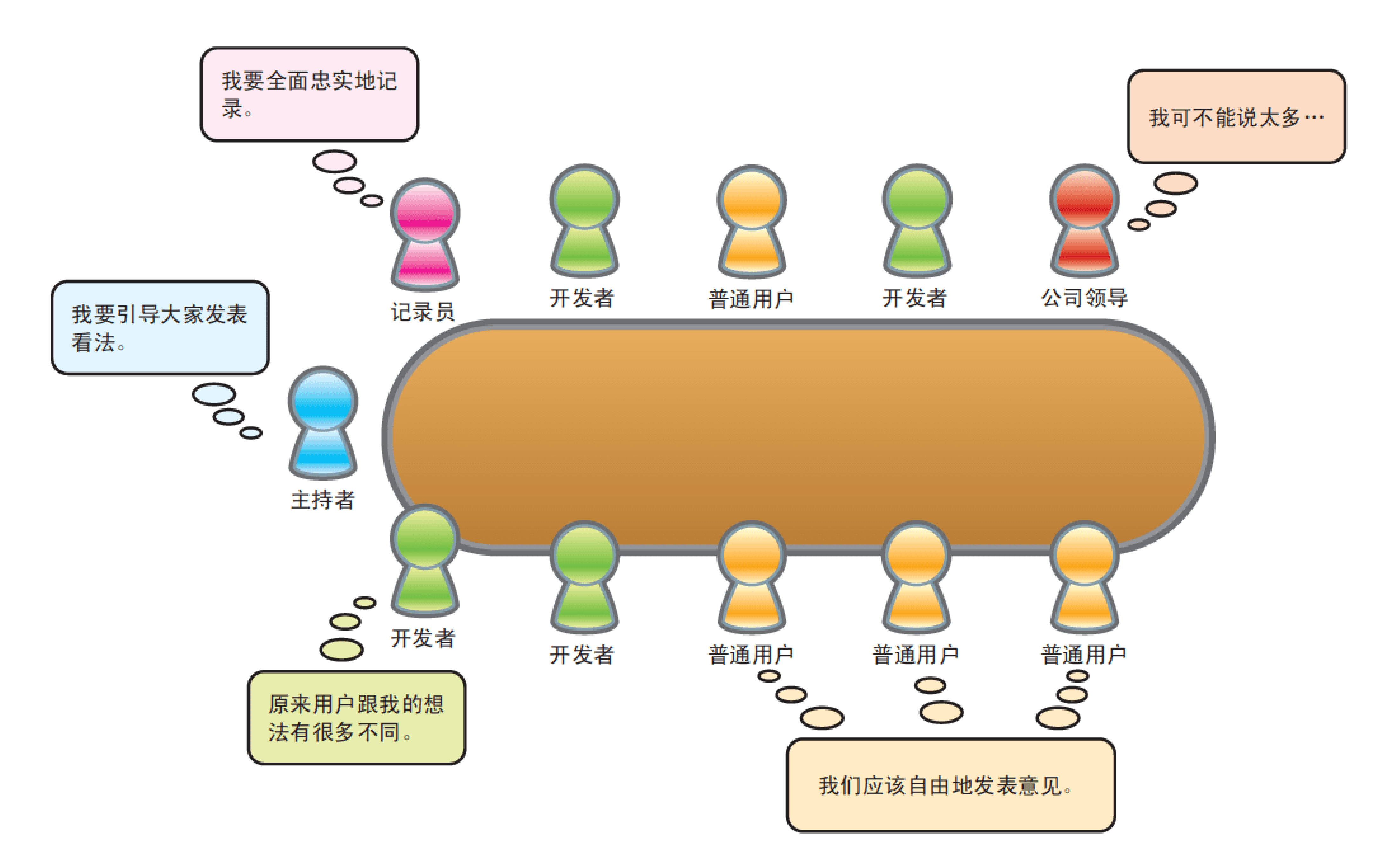
在观察和提问的过程中，如果你发现很难同时进行记录，也可以考虑对过程进行摄像或者录音。这样就算结束之后，你也能研究各种细节。但是当有些人在知道自己被摄像的时候，他们的行为可能会跟正常状态有所不同，所以你应该通过各种方法减小这种影响。

观察和提问需要我们投入更多的时间和精力，而且可能会收集过多的数据（想想摄像得到的数据）。而且研究这些数据也会耗费你更多的时间。

集体讨论

刚才提到的用户访谈通常都是一对一的，只能知道一个人的看法。你也可以召集若干名用户聚在一起进行讨论。

一般这需要一位主持人，负责组织整个讨论，而且要保证讨论不超出预定的范围。他还应该负责引导大家专注于某个问题，或者转换问题。



集体的专题讨论适合于了解多数人的意见，也能突出有冲突、不一致的观点。让设计人员和用户都注意到自己的观点可能与其他人并不相同，这是非常有益的。

不过，你要特别注意避免一个或几个人垄断谈话内容、或者主导讨论的情况发生。尤其是那些有控制权、职位高或者有影响力的人，他们养尊处优，总觉得自己是最重要的。你应该尽可能保证所有的讨论者都有相等的机会充分地发表意见。

另一方面，有些人可能天生就很内向，不善言辞。与之相反，另一些人看到领导在场就特别亢奋，有意拔高自己的形象。还有些人有领导在场的时候考虑过多，不敢提意见。这都不利于集体讨论。

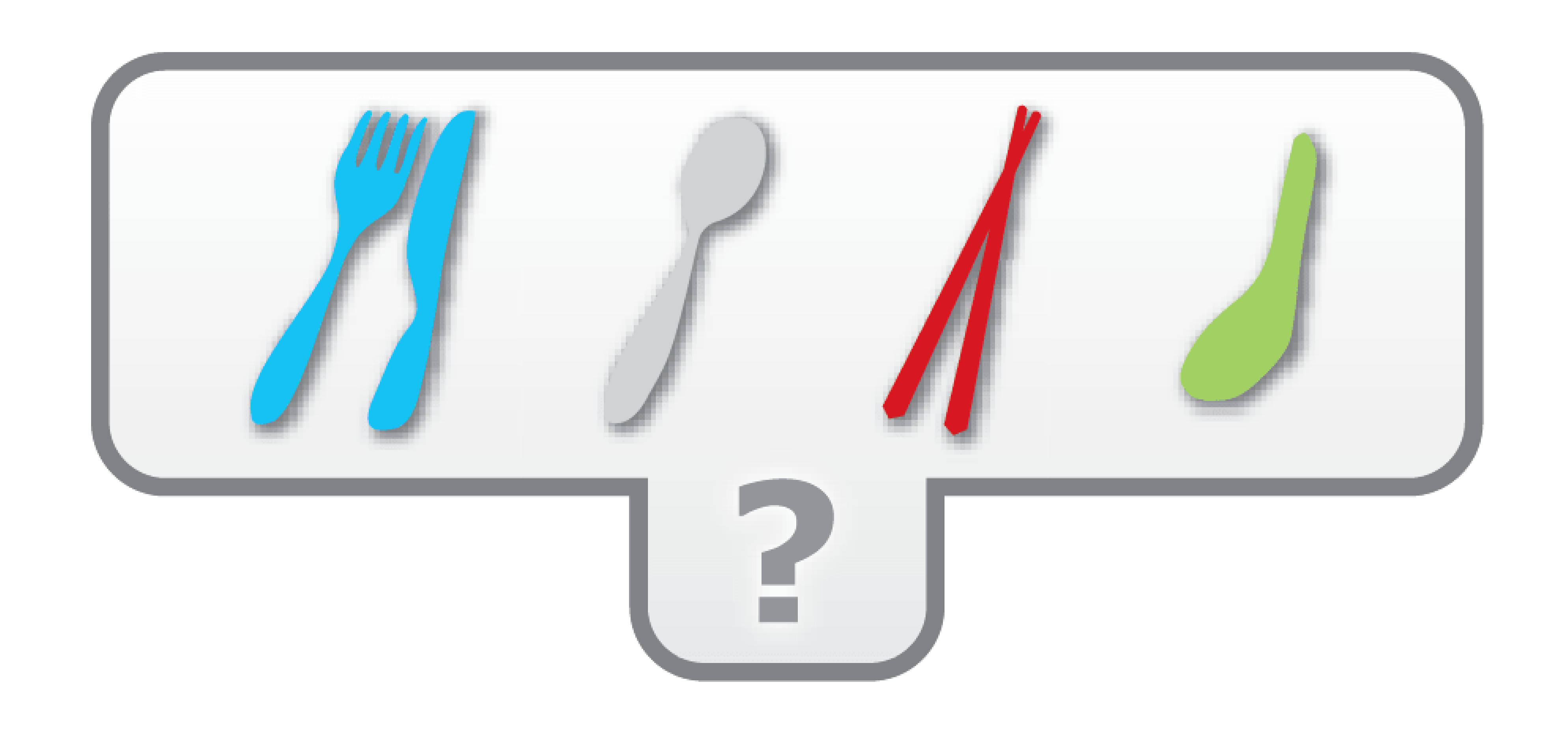


我觉得，我们的产品一点毛病都没有。
当然你们也可以提出你们的意见……

针对这一问题，你也可以通过以下方式来避免：让参与讨论的用户各自使用一台电脑进行匿名在线会议（例如 NetMeeting 等网络会议软件，甚至是 QQ、MSN）。“藏”在计算机后面键入谈话内容，避免了正面的对话，谁也不知道谁是谁。这样就能使所有参加讨论的用户都能有相对平均的机会提出看法。

另外需要指出的是，因为这种会议无法在操作现场进行，而且讨论的情况和实际使用的情况往往有偏差，所以集体讨论的结果只具参考价值。

2.3.2 筷子、刀叉和勺，哪种餐具最好？



这个问题的答案因人而异，也需要根据不同的情况来决定。

首先是看你最习惯使用哪种餐具。此外不同的食物也要有不同的选择。而就算是一种食物，你也可能需要好几种餐具。例如用勺子盛汤，用筷子捞里面的排骨。

上面说的种种数据搜集方式也是一样，你需要根据自己的能力，还有项目本身的特点来选择各种方式。另外，它也取决于你在开发周期的什么阶段。比方说，在项目开始时，你可能找不到什么问题来询问用户，所以最好的办法是先通过访谈来发掘各种问题，而不是问卷调查。

同时，你的选择也会受到可用资源的影响。例如，如果要在全国范围内进行问卷调查，就必须要有足够的时间和资金，之后还得有能力去整理和分析海量的结果。如果只有三周的时间，又没有什么相关的经验，这个方法就不太合适。再比方说，如果用户分散在全国（甚至全世界）各地的话，也不适合把他们召集起来进行集体讨论。

	适用情形	优点	缺点
问卷调查	回答特定的问题	使用资源很少，调查人数很多	问卷设计很关键，回答率可能不高，也可能答非所问
用户访谈	发掘和深入研究问题	必要时可引导被访者。能促进开发人员和用户的沟通	很耗时，人为环境可能使被访者感觉不自在
观察和提问	理解用户活动的环境	观察实际工作，能够提供很多的细节	非常耗时，数据量巨大
集体讨论	搜集多方观点	能突出一致和不一致的观点。能促进开发人员和用户的沟通	讨论可能由少数人主导，讨论和实际的情况往往有偏差

也许我们看看具体情况会更容易理解。

比如说，现在有个小型广告公司的在线管理系统需要升级。这是一个正在运行的 Web-based 系统，用于管理公司内部的考勤、工作文档以及流程等方面。公司的员工对它比较满意，但是系统有些过时。

那么我们看看可以通过哪些数据搜集的方法来获得用户的需求信息。因为这是一个小型的广告公司，所以用户的数量并不多。所以与用户进行访谈要比问卷调查更合适，访谈也能获得更丰富的数据，开发人员也有机会与用户会面。“观察和提问”是必要的，这能帮助你理解新、旧系统的使用环境。

因此，我们应该首先与主要用户进行一系列访谈，了解已有系统的优缺点，再进行简短的观察并提问，从而理解系统的应用环境。

而如果是要开发一个面向年轻人的时尚电子商务网站，就得换种思路。因为用户是分散的，与他们一一会面并不现实，而且最重要的是你没办法很准确地定义用户。所以你可以先进行大面积的电子问卷调查，看看人们希望看到的电子商务网站是个什么样子。然后你可以从中挑选出一部分有代表性的用户进行访谈，或者组织一些人进行集体讨论，以作为问卷调查的补充。

此外，找出相似或者竞争对手的网站，对它们进行评估，也有助于提供有用的信息（这是个很好的办法，你能免费得到一个不错的参照对象）。

2.3.3 数据搜集活动中一些要注意的地方

这些数据搜集活动并不难，你所需要付出的只是一些耐心和时间。另外，必要的策划也是需要的，同时也要搞清楚你的目标——一切都是为了理解用户和他们的任务。这需要使用科学的方法，而不是率性为之。

一切的重点都是为了搞清楚用户需要什么

了解用户的需要，是为了确定需求。有了需求，后面的设计工作就有章可循，而不会像无头苍蝇一样乱撞。你研究的是用户的行为，他们的使用环境，还有现有的一些支持工具。如果能考虑竞争对手的产品的话，效果会更好。

要考虑所有的用户类别



什么？供应商不知道我们的新系统怎么用？

你是否能保证搜集到了所有用户的观点？很可能一不小心就会忽略某些用户。曾经有一个公司升级自己的 Web-based 企业管理系统，设计人员非常谨慎，而且也十分注重用户的感受。他们让所有职员、管理者甚至仓库管理员都参与了开发过程。但是系统正式运行后，生产效率却比以前降低了一半。问题出在哪里？他们确实考虑到了公司内部的所有用户，但却忽略了不是公司成员的供应商们。没有人考虑过供应商该如何使用系统，但很明显他们在使用上出现了问题。

每个用户类别只派一位代表参与是不充分的

不光用户的类别要考虑全面，就算是在单个的类别中，也需要征求多位用户的意见。那些大型的产品或服务尤其如此。对于如何工作、如何交互，每个人都会有不同的看法。就算只有一个用户类别，但如果你只访谈或者观察一个用户，肯定没什么说服力。

打“组合拳”，不要单一套路

各种数据搜集的方法都是从不同的角度捕捉不同的信息。它们谁也取代不了谁。这就好像练武术只练腿上功夫是不够的一样。



比如说你可以通过观察和提问来理解用户执行任务的具体环境，然后通过访谈来了解特定的用户，进行问卷调查获取更广泛的意见，最后通过集体讨论来达成最终的一致意见。

在可能的情况下，先小规模试验

对于问卷调查来说，预先的小规模试验是非常重要的，你需要检验自己设计的问卷有没有问题。集体讨论也是同样的道理，那么多人齐聚一

堂，难免可能会出现局面失控。你可以通过试验发现大概会出现什么情况，如何处理，从而避免意想不到的情况发生。小规模试验能确保你的数据搜集活动达到预期的目标。

不过，一般小规模试验后很可能会进行修改，这就表明这种试验是不正规的。所以，这个过程中获得的数据不能与其他方式获得的数据等同看待。

如果得到的需求信息太多

很多情况下，你获得的需求信息可能有几百条、上千条，甚至更多。为了防止用户信息过于杂乱导致增加工作量，同时又要保证需求的可衡量性，你可以采取以下的步骤来了解需求：

1. 请用户列出他们希望通过产品完成的各种目标。
2. 让他们笼统地列出达到目标的各种需求（并不需要多么精确）。
3. 按你自己的理解，把这些需求再转述给用户，请他们来判断是否正确。
4. 鼓励用户以更开放的思路提出更多需求。
5. 你提供几个需求类别，让用户将所有需求“对号入座”。
6. 列出各项需求的衡量标准。对那些不易衡量的需求，进行修改或删除。

记录数据同样很重要

在面对面的数据搜集过程中，例如观察、访谈和集体讨论，你主要可以通过摄像、录音和记笔记等方式来记录必要的数据。摄像和录音的优势是非常准确，但得到的数据量太大。记笔记则需要保证即时、准确，不能有偏向性。

2.3.4 解释与分析数据—— 一道填空题

到现在，你通过问卷、观察、访谈以及讨论已经得到了一些信息。你知道它们对开发工作是有帮助的。但是，如果只是拿着厚厚的一沓记录（也许还有视频和音频），却找不到一个办法把它

们组织起来的话，导致的结果只可能是数据的堆砌。它们就在那里，你却没有任何头绪。

所以你需要有一个清晰的思路来整理这些信息。或者说，你得通过解释这些数据来组织和陈述用户的需求。

我建议你在完成了某一个数据搜集活动之后，马上就开始数据的解释与分析工作。越早越好。这个时候你的记忆还很清晰，能想起用户说起的每一句话，他的每个动作。细节非常重要。如果你想搁一搁，到明天再说，那么哪怕只隔一个晚上你就能忘记很多东西。

那么如何解释数据呢？刚才说到这个阶段的目的是要组织和陈述需求。关于这一点有很多办法，但最好的办法就是使用一套比较完整的模板，从而把需求文档化。你可以根据模板的各个项目来理解需求数据，这就像一道阅读理解的填空题，你所要做的只是看看哪里该填些什么。

大西洋系统协会公司（Atlantic Systems Guild Inc.）所提供的 Volere 需求记录卡模板就非常实用。它主要包括下面这些内容（我作了一些小小的修改）：



任何需求都无非以上几点，所以一看就很清楚。在这种时候，尽可能多地召集相关设计人员一起讨论是个很好的办法。因为每个人的思路不一样，有些人可能会比你从中发现更多的东西，这样你就能够得到关于数据的各种看法。

把这道填空题做完，你就能得到一张需求记录卡。之前获取的各种需求信息都能够通过这种

方法来统一归纳。把所有的需求记录卡集中到一起，你就能建立一套完整的需求记录。

有一套综合网络管理系统，它能针对电信公司的系统、用户、主机等多方面以 Web-based 方式进行管理。在开发的过程中也许会存在下面这样一张需求记录卡：



这张需求记录卡属于故障告警类中的一张。内容描述了故障发生的原因、提出需求的用户，以及满足需求所需要达到的目标等等。

这个公司把满意度的级别分为非常满意、满意、较满意、不满意和极为不满意五种。

你可以看到，与这项需求相关的是“弹出窗口的 UI 设计规范”需求。也就是说，当系统告警时，告警界面需要遵循相关的文档。此外，用户对系统没有其他需求与之相冲突。

OK，一道填空题，目的达到了：通过这张记录卡，你就能明白无误地把握用户对这一方面的需求。这非常明确。当你把这些需求收集在一起，拿订书机订起来，就能得到一本相当不错的需求记录汇总了。

2.4 归根结底，我们要了解任务

事情马上就完，但还差最后一步。

我能想象这种情况：用户看到你给他的一沓需求记录卡，他仔细地读完并且点头说：“想当

不错，这就是我们的需求。”但是，他是不是真的理解了？

其实，他看到的只是一些要求，以及你承诺的相应举措。也许他很满意，但其实他看到的并不是产品，所以也不会形成明确的概念。

这就好比农村来的小波波说他很无聊，你承诺周末带他去迪斯尼游乐场。他可能猜到会很好玩，但从没去过游乐场的波波并不知道那是个什么地方。

这种情况下你会怎么做？你可能会说：“那里棒极了！有小丑和摩天轮，你能吃到冰激凌和棉花糖，还可以坐过山车。”

就是这样。你需要向他描述那里会发生的一切，让他脑海里产生具体的印象。然后波波就会缠着你带他去了。对用户来说也是如此，你需要用具体的例子来向他们解释如何使用产品，会发生些什么事情。这一阶段我们称之为“任务描述”。

所谓“任务描述”，就是通过某种方式把具体的任务以用户能理解的方式详细地表现出来。这样，他们就能一目了然地明白正在开发的产品大概是个什么样子，使用起来会遇到什么情况，有什么问题。任务描述最终的目的是能够让你借助这些找出确切无误的需求，也就是整个标识需求过程的最后一道关。

一般来说，任务描述不应涉及任何与技术实施相关的内容。也就是说，在你描述任务的时候，那些系统的运行平台、信息的存储方式等等在“背后”的东西都应该抛开。用户们不关心这些问题，他们只会注意那些展现在他们面前的东西：界面如何操作，任务如何完成，等等。说太多技术上的内容，几乎就等于你在絮絮叨叨一些毫无意义的东西。



另外，通过任务描述，你也可以对用户的任务进行分析。一个优秀的任务描述可以回答几个相当详细的问题：

- 使用该产品或服务的人都会做什么任务？
- 哪些任务是常用的，哪些则很少用到？
- 哪些任务最重要，哪些最不重要？
- 每个任务的步骤和结果是什么？
- 每个任务的信息从哪里获得？是不是由系统提示出来？
- 哪些人做哪些任务？
- 完成任务的时候都会使用到什么工具？
- 执行每个任务时人们会有什么问题？什么类型的错误比较常见？是什么引起的这些问题和错误？

通过回答这些问题，便于你更为全面地了解任务。这是非常重要的。

任务描述有很多种不同的形式，我在这里主要介绍其中的两种：场景和流程图。这两种方法并不相互排斥，正相反，通常很多人都是结合两者来使用。因为它们都各有所长。

其实我想你对它们也不会陌生。

2.4.1 用讲故事的方法来描述任务

“讲故事”指的就是使用“场景”来描述任务，它是一种非正式的叙述方式，但从中你可以发掘出用户执行任务的环境，以及他们的需求。

“场景”描述的是用户完成任务的具体情节。这种“讲故事”的方法非常便于理解，使用的是用户的词汇，轻松友好。用户很容易就能接受这种方式，从而理解场景说明，也因此能够参与开发过程。

设想你马上要开发某个公司的人事管理系统，于是应邀与用户会面。他们负责公司的人员招聘工作。当你走进办公室时，负责人苏瑞跟你打招呼。你觉得她会怎么说？

“是的，求职申请都是寄到我们这里。有时候高峰期我们每天可能会收到约 30 到 40 份简历。朱迪负责浏览简历，检查是否完整，也就是看看它是否包含所有需要的材料。在处理求职申请之前，我们需要检查相关的文凭、证书以及工作经历证明。根据这个初步审查的结果，我们再把这些求职申请转交给……”

“讲故事”是人们解释自己做什么或者如何执行某个任务的最自然的方式，它便于所有人理解。这类“故事”的焦点通常是用户希望达到的目标。通过理解用户为什么这么做、想达到什么目的，

你就能专注于用户的活动，而不是单纯的技术。换句话说，这个“故事”的情节仅仅是指任务（他们的目的）和操作（他们的行为），而不需牵扯任何的技术问题。

在任务描述中，使用的故事或情节可以是真实的，也可以是虚构的；可以是关于使用当前存在的系统的情况，也可以是想象中的理想情况；可以来源于用户本身，也可以由你自己编写出来。

这种方法的关键是要使这些故事和情节具有代表性，从而可以作为设计的参考。

一位曾去某个书店买书的用户可能会提出以下场景：

上周日我去 ×× 书店想买一本关于交互设计的书。我比较中意其中一本，但书名记不清了。但我知道书的作者是付利民，他好像是美国 eBay 的用户体验研究经理。

书店里陈列的书太多，于是我准备使用旁边的一台电脑搜索一下，看看能否找到。如果还不行的话就只有去找售货员了。我走近电脑，发现有一些人正在使用它们。5 分钟之后终于轮到我了。我看到计算机屏幕上打开了一个网页，上面显示“×× 书店图书查询系统”，但需要我输入用户名和口令。我不知道用户名和口令，但不输入它们我就无法进入系统。这时候我看到计算机显示器旁边贴了一张纸条，上面写着：“用户名：user。密码：123”。我尝试着输入，结果成功了。

系统确认了口令后，显示可以根据作者、领域类别、书名或版号查询。我发现不能同时结合两者进行检索，比如输入作者名字和“计算机类图书”。我不知道那本书叫什么，而如果搜索“计算机类图书”又肯定会得到一大堆结果。所以我选择搜索作者名。过了约 20 秒，系统返回了结果，提示我没有找到付利民写的书，但也列出了一些最接近的匹配书目。

在浏览了清单后，我发现把作者的名字弄错了，应该是傅利民，而不是付利民。于是我选择了想要的书目，系统随后显示了它的陈列位置。

一个完整的故事描述包括人物、目标、现状、环境、步骤、策略、感情等多方面的因素。分离这些因素对于任务的分析是很有帮助的。下面是对于上述故事的分解：

- **人物**：“我”，一个想购书的人。
- **目标**：购买一本关于交互设计的书，但只知道作者名字。
- **现状**：周末，书店有些拥挤。
- **环境**：书店，若干查询终端，另外售货员也可以提供帮助。
- **步骤**：先浏览陈列架，然后使用查询终端，登录系统，查询图书，获得书籍信息。
- **策略**：如果使用终端找到了想要的书籍，则直接取书购买；如果无法找到，则向售货员

寻求帮助。

- **感情**：得到了想要的结果，但对系统需要登录感到不理解，而且对不灵活的检索方法感到无奈。但是很庆幸最终系统给出了近似的搜索结果。

从这个关于已有系统的场景说明中，我们注意到了一些事项：

- 正确输入作者姓名的重要性；
- 用户对强制进行没有意义的登录感到莫名其妙；
- 应提供更为灵活的检索方法；
- 检索速度应该努力提高以免用户排队；
- 在匹配不成功时，应给出相近的检索结果。

在设计新的图书查询系统时，这些都是可以改进的地方。这个场景描述了图书查询系统的一个（很可能是普遍的）用法，即根据作者的姓名检索一本不知名的书。

从上述的故事讲述中可以看出，场景描述这种方法比较生动地叙述了一个典型用户使用系统的完整过程。故事讲述包括很多细节。除了系统本身的功能之外，故事描述还涉及了很多环境和辅助因素，例如有没有售货员、顾客是否需要排队等等。这些都与系统的设计密切相关，应该考虑进来。

描述已有的场景有助于你决定新的场景。你能从中搜集到很多有用的信息,以便建立新的需求。现在我们来看看如何借助现有活动的场景说明来找出关于崭新应用的需求。

比如说你想选购一张 Hi-Fi 发烧音碟，一般情况下会怎么做呢？我编写了一份场景的例子，它并不一定能代表通常的情况，但我估计也差不了太多。

我是个喜欢音乐的人，但是个外行。我不知道门德尔松和施特劳斯之间的区别。如果要买张好的发烧碟，我不太了解该怎么做——你知道好的原版唱片对我来说并不便宜。我可能会上网搜集一些关于发烧碟的音乐评论，不同的风格、质量，当然还有售价。顺利的话，从这些初步调查中，我会发现一些比较中意的唱片。

下一个步骤就是去音像店，但是似乎有些音像制品的价格没有一个明确的标准。我会选择我喜欢的音乐家的作品，从架上拿下来欣赏一下封套。但我仍然不确定它的质量是否真的很好。如果音像店能够提供试听，我会很高兴，但是只有很少的音像店提供试听服务，就算有，也就那么几张摆在那里——*似乎此举是为了展示耳机的质量而不是唱片*。如果能够试听，我通常就能立即决定是否购买它。我不想花几百元买回去后发现不是我想要的。

这个场景表明，这项任务大体上分为两个阶段：一是研究市面上各种发烧音碟，二是获得实际的体验。前一个阶段是获取一些实际的信息，例如在网上搜寻资料。但搜集的过程比较零散，而且正如我们所看到的，即使是同一张唱片，不同音像店的标价也可能有所不同。在后一个阶段中，试听是非常重要的，可以说是一个关键的步骤。但是正如上面所看到的，大多数音像店都不提供试听服务，少数有的几家也只是装装样子，提供的唱片屈指可数。

根据以上说明，我提出了以下场景，描述了新式的“一站式 Hi-Fi 发烧音碟在线商城”。它能够满足一些用户在这方面的部分需求。

我想买一张好的发烧碟，所以来到了“一站式 Hi-Fi 发烧音碟在线商城”。这是一个网站，我可以直接选择我想要的唱片，也可以在没有明确需求的情况下看看有什么比较合适的。

网站里面提供了市面上绝大部分 Hi-Fi 唱片的资料，包括封套设计、作曲家、演奏家、唱片风格以及一些专业和非专业的评论。资料还包括唱片在网站上的售价。我能够灵活地根据价格、作者或音乐流派进行检索，还能够阅读所有资料。我很感兴趣，花了半个小时浏览各种资料，然后选定了几张可能会购买的唱片。

每张唱片的图片下方都有一个“试听”按钮。我可以随时点击，对专辑的大部分曲目进行试听。曲目都是一些片段，但是音质很清晰，能够把握细节。最后我选定了一张小约翰·施特劳斯的《维也纳的浪漫》，这正是我想要的，我还从资料上了解到这种音乐风格叫做“波尔卡舞曲”。

可以看到，这个网站支持前一个场景中提到的研究过程和重要的试听过程。它应该能满足大部分用户的需求。当然这只是一个初步设想的场景，还需要通过讨论和进一步的调查做出改进。

2.4.2　用流程图来描述任务

流程图相信你早已见过。在任务描述中使用的流程图，同样关注的是用户目标，但它的重点是用户与产品的交互，而不只是任务本身。

流程图的作用是分析基本原理，了解人们想要达到什么目标，然后明确地表现出应该如何达到这些目标。它描述了现有任务的执行情况。以此为基础，你就能建立新的需求或者设计新任务。

简单来说，流程图就是用具体的流程框图来明确表示任务执行的完整过程。

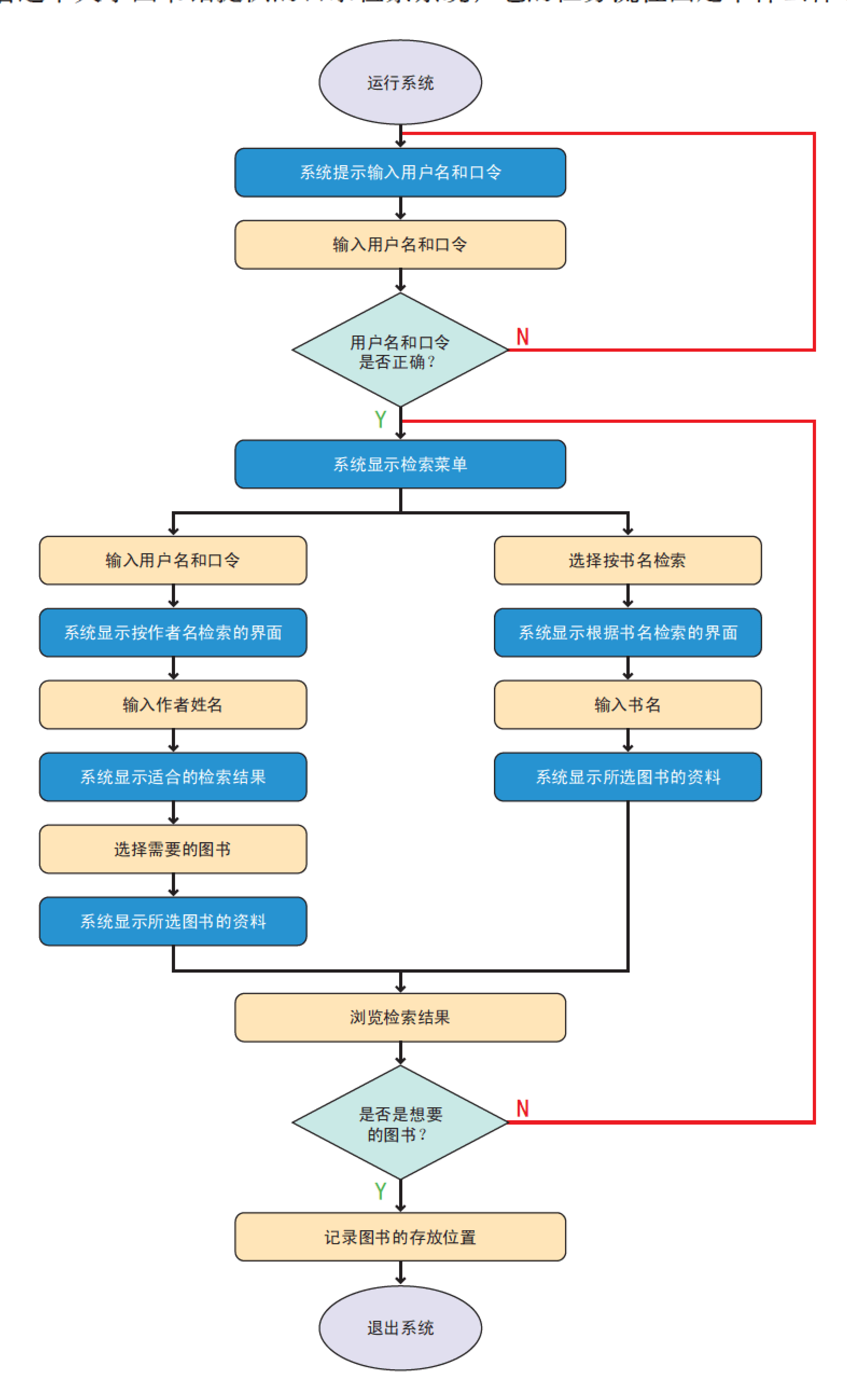
举个例子。你在某个大学图书馆使用它的目录检索系统。这个系统允许用户检索详细的图书资料，例如根据作者或主题检索图书，然后找到图书存放的位置进行借阅。

需要知道的是，不同的用户完成某项任务的具体方式并不完全一样。而且他们在不同的情况下也可能会随时调整自己的步骤或策略。在图书馆这个例子中，用户会根据自己是否知道书

名来选择用何种检索方式查询图书。也就是说有些用户会直接检索书名，有些用户会输入作者名来查询。不同的情况下步骤并不一样。这些所有可能发生的步骤都需要表现出来。

流程图能够完整地综合表达不同用户在不同的条件下完成某项任务所可能采取的不同步骤和策略选择的情况。它描述的是任务的整个过程，以及用户在过程中的决策。

我们来看看这个大学图书馆提供的目录检索系统，它的任务流程图是个什么样子：



在上面的流程图中，蓝色的步骤表示系统的响应，黄色的步骤表示用户的操作。我们可以看出，流程图能够使任务的整个操作步骤一目了然。你能够知道用户在完成这一任务时可能的步骤和策略，以及系统应对其操作所做出的响应。这便于你清晰地认识任务。

2.4.3　理想和现实的差距

记住，不要在任务描述中提到关于技术方面的问题。只有这样，你才能给用户不受约束的最大的思维空间，去设想他们最想要的产品。

如果用户在关注任务的同时，老是不安地问你“这能不能实现出来？”、“那个实施起来有问题么？”，他们的思维肯定会受到限制，或者不自觉地为技术让步。这样出来的结果要么就不能令他们满意，要么就可能存在着不同程度的可用性问题。

我们需要分两方面来讨论这个问题：

守旧的用户

在很多情况下，要想完全抛开已有的思维定式是很困难的事情。我总是想到《印第安纳·琼斯》里面那个场景：在故事里，悬崖上有一条悬空的路，但由于视觉的关系根本看不见，只有万丈深渊。没有人敢走过去，直到某个人扬了一把沙，沙子落在透明的路上产生了清晰的实地感觉，才能有勇气踏上去。

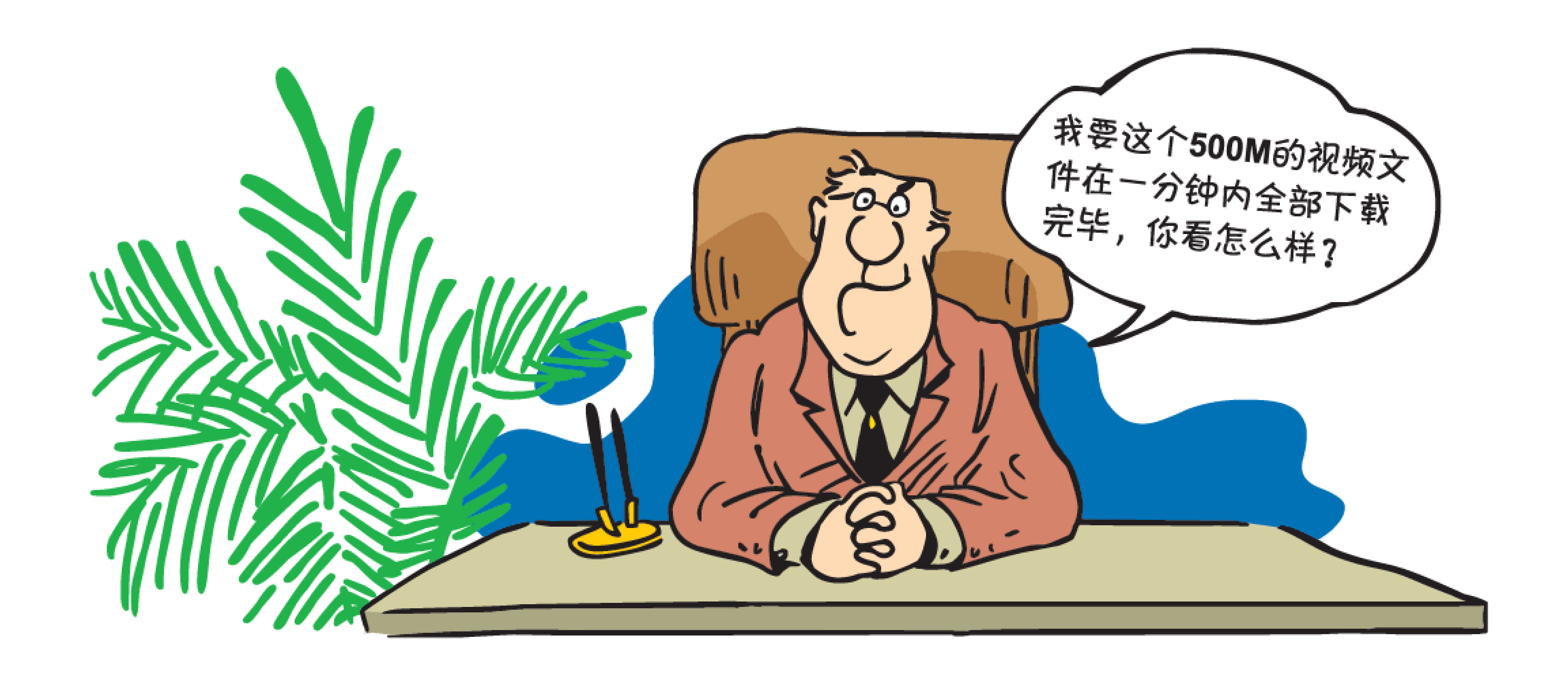
这就是思维的定式。如果没有沙子铺成的平面，我就无法相信这是一条路。设计也是一样，用户总是会想到现有的某些相似产品是如何实现的，他们会下意识地往那个方向去靠拢。你想让他们创造性地给你一个新需求，并不很容易。

例如你要设计某个网站，用户总是会很自然地联想到现有网站的运行方式，或者一些其他网站的类似功能。如果是这种情况，只要他们关注的是完成任务，而不是技术的局限，那么就没问题。但如果他们总是去考虑技术是否能支持他们达到目标，别让这些束缚他们。让他们大胆去憧憬，只有这样你才能得到真正的用户需求。

过分的用户

毫无疑问，理想和现实总是有差距。技术局限总是存在的，用户们大胆的憧憬往往也会与实施方面的各种局限相冲突。

由于不了解相关的技术常识，有些用户又是另一种极端。例如有些人的要求就很过分：



你并不能说他们是恶意的，或者无理取闹。他们只是不明白现实的限制。

要以用户为中心，但又无法一味迁就用户。看来，做一个好的设计师真的很不容易。

妥协还是抗争？

所以在你决策设计方案的时候，往往需要对用户的要求和实施局限这两方面加以权衡，并不得不加以妥协。

虽然这种妥协是必要的，但是如果对技术的妥协过多，偏离了用户所希望的设计方向，就必然会出现一些可用性问题。当你对他们说："我不是神灯，你提到的 10 个要求里面我只能完成 1 个"的时候，他们肯定会很不高兴。所以在这个地方我们应当遵循一条原则，那就是：

> 对用户理想的完成任务方式的支持 **大于** 对技术局限的迁就。

也就是说，如果某种技术局限可能会严重影响产品的可用性，那么你就应该努力突破这种技术局限。



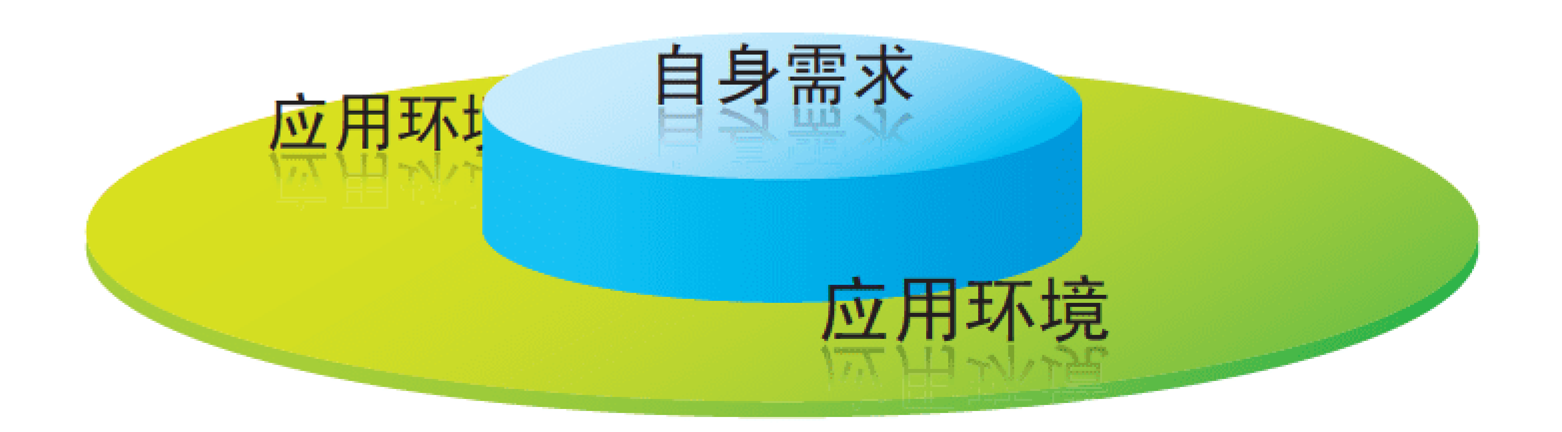
有时候事实就是这样……

2.5　环境？什么是环境？

我们在上文中经常提到一个词："环境"。

不，这不是生态环境，也不是自然环境，我们谈的是交互设计中的"环境"，也就是在任务过程中用户所处的环境，或者说是产品或服务的应用环境。

每个人都关注他自己的目标和愿望，这很正常。但他们还应该考虑到，在周围环境中可能会有一些其他的事情，这些事情可能会影响到用户使用产品的结果。



看看生活中常见的汽车报警器。开发报警器的人想设计一个能在汽车被盗或被破坏时发出警报的工具。而作为购买者，他们只是在考虑如何保护自己的车。从这两个方面来说，他们都没有问题。

但是两者都没有考虑到环境的因素。环境是可能其他许多人都有同样的报警器，而且除了非法破坏之外，还有许多事件都可能会触发报警器。

当某个报警器在附近响起来的时候，你会发现很难辨别出到底是谁的车在响，也不知道是不是

真的有人偷车或者故意破坏。一般来说大部分情况下都不是。我们经常听到突然响起的警报声，在你探头往窗外查看的时候——仍然是宁静的夏日，狗在打着呵欠。

所以，你也看到了，这里面似乎有些问题。想法很合理，产品质量也没有问题，但是产品好像并没有提供多大价值……其实并不是想法本身有瑕疵，而是因为设计者和购买者都没有考虑到更大的环境。

当你没有以用户为中心，而只是关注技术的时候，就如同一片漆黑的舞台，只有一台计算机上面打着聚光灯。我们只能看到人走到聚光灯下，使用计算机，然后离开，再次消失在黑暗中。他或她为什么来使用计算机，不知道；使用后会发生什么，不清楚。在这种情况下，你只能知道“这个产品应该做什么”，但其他的东西都无法弄清。



而正确看待这一问题的方式是，舞台应该有充分的光源，照亮了其他一些角落。信息、通信和工作在舞台上流动，我们能清楚地看到它们来自何方，去向哪里，在那些区域当中会发生什么没有预期到的问题。也就是说，当你理解了产品和服务的应用环境后，就会在关注“这个产品应该做什么”的同时，还关注“它如何完成这些工作”以及“哪种产品或服务会（更好地）支持完成这些工作”等问题。

所以说，环境是十分重要的。在了解用户需求的过程中，你必须时刻关注用户所处的环境。而为了理解这种应用环境，就必须研究它。这意味着你要与具有代表性的用户交谈，观察他们并且分析由此收集到的数据。

2.6 做个小结

这一章所谈的是交互设计的第一个内容：“理解用户的需要，从而建立明确的需求”。 正确建立需求对于交互产品的成功是至关重要的。

首先你了解到“用户”和“客户”之间的区别。“以用户为中心”的主要对象是用户，也就是真正接触和使用我们开发的产品的人。用户具有人类的共同特性，但对于产品来说，他们又彼此不同。让开发的产品能适用于“每个人”是不可能的。

然后我们学习了如何建造自己的“阿童木”——用户模型。

在了解用户需求的时候，你先得了解用户的“需要”。如果大家都不知道需要什么，就只能研究现有或过去的相似产品，从中得到启发。

需求也有很多类型，包括功能需求、用户需求、数据需求、环境需求和可用性需求等。每个产品都存在这些不同的需求，所以不要顾此失彼。

此外在收集需求的过程中，还需要强调需求的“可衡量性”。对每个需求来说，你都要知道做到什么地步才算是达到了目标。对于那些完全不能衡量的需求，别在上面浪费时间。

要想得到需求的信息，最常用的方法有问卷调查、用户访谈、观察和集体讨论。这些技术或方法都各有其优缺点，所以你可以根据具体的情况综合选用、取长补短。

我想你同时也了解到了产品应用环境的重要性。在得到了来自于用户的数据和信息后，你可以使用需求模板把它们整理归纳为具体、明确的需求记录。

最后我们了解到，把用户的任务通过“讲故事”或流程图的方式描述出来，能够帮助用户明确说明实际的工作，也可用于展望新产品的使用。

以上就是确定需求这一步骤所需要弄明白的知识。其实这些并不难理解，我们要做的就是多和用户接触，通过一些科学的方式和用户沟通，从他们的言行举止中了解他们对产品的看法，从而将其整理为规范的需求文档。

我必须承认你干得很棒。



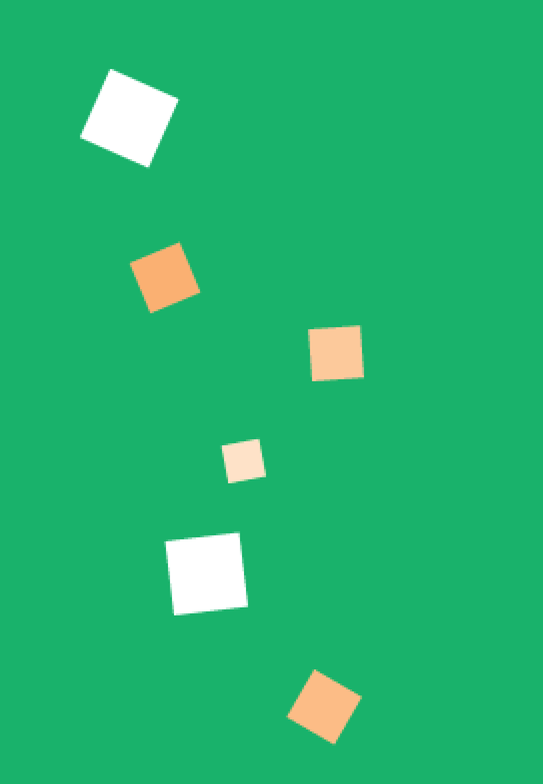
有了需求，我们才能知道具体该如何去做，这就是所谓的“有的放矢”。盲目地以自己的需求或片面的理解去为用户“设计”需求是不明智的，最终出来的成果很可能经不起考验。如果最终满足不了用户，必然也就无法获得经济上的收益。赚不了钱的产品还不如不开发。所以，要想开发优秀的产品，第一步就是要明白自己该怎么做。

那么现在你应该明白了“用户想要什么”和“我们要做什么”。掌握了这些之后，我们该进入哪一个步骤呢？就这么直接开始正式的设计工作无疑是鲁莽的，因为你的思路还没有获得验证。产品的开发需要耗费大量的时间、物力以及财力，如果在最终发现与用户的期望相差太远，这一定是任何人都不愿意看到的。为了避免这种情况的发生，你就必须在正式开始设计之前对自己的思路进行验证。你需要先提出你认为能够满足用户需求的构思，然后把这些构思展示给用户，让他们来评判是否合适。

展示构思最经济有效的方法就是设计和制作原型。

第三章

设计方案和制作原型



我们甚至可以用钢笔、铅笔、橡皮、便条纸和名片来当作原型进行方案的讨论。

这是确实存在的专业技术！它叫做“PICTIVE”。

我们之前说过，用户往往不能准确地描述自己的需要。这不是说他们笨或傻，只是因为他们作为用户并不具备开发者所具有的认识。



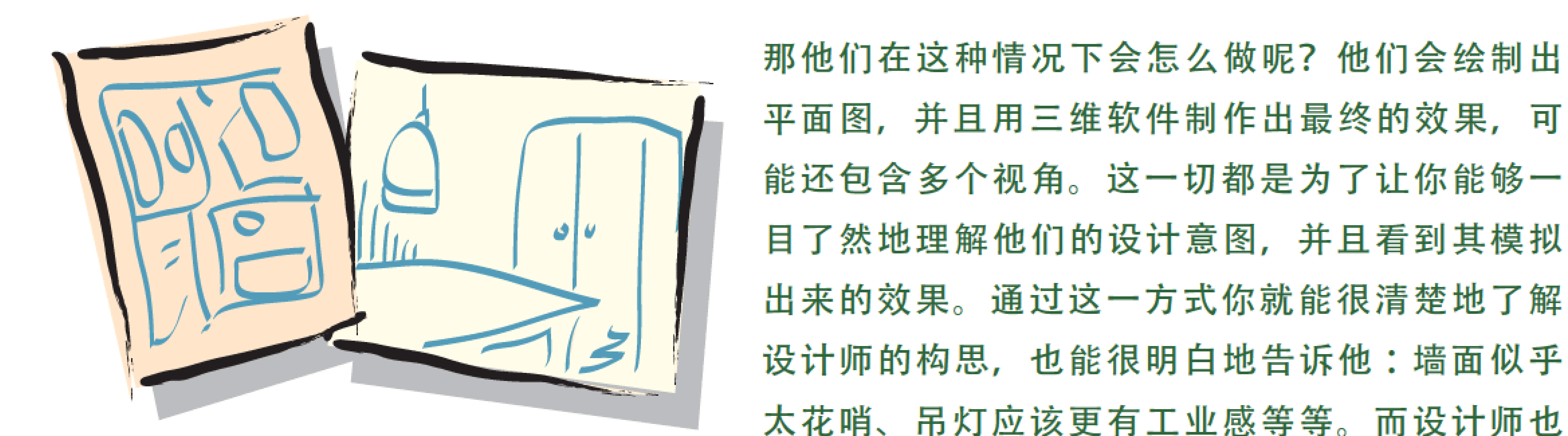
而一旦他们看到或尝试了某些事物之后，就能立即知道自己需要什么，不需要什么。他们可以指着某些地方告诉你 ：“这玩意让我迷惑”、“那个很讨厌”或者“你说我能把它打印出来，但是打印按钮在哪里？”

但是，如果等到一切都做好了之后再拿去让他们指出问题就有点晚了。你已经投入了很多时间、精力和金钱，万一用户并不接受，难道就意味着一切都要从头来过？

这里面似乎仍差点什么东西。这就如同我们必须去找把梯子才能修吊灯，不是吗？

3.1 为什么，以及怎么做

在装修房子的时候，装修设计师只口头叙述房间如何布置是不够的。我们中的大部分人可能都不能根据他的描述形成鲜明的印象。设计师也不可能说“好，等着”然后把房子装修好，再打开门让我们进去。万一我们不满意怎么办？这所有的木料、瓷砖、吊顶包括灯具、线路可能全部得拆掉重来。没有人会这么蛮干。



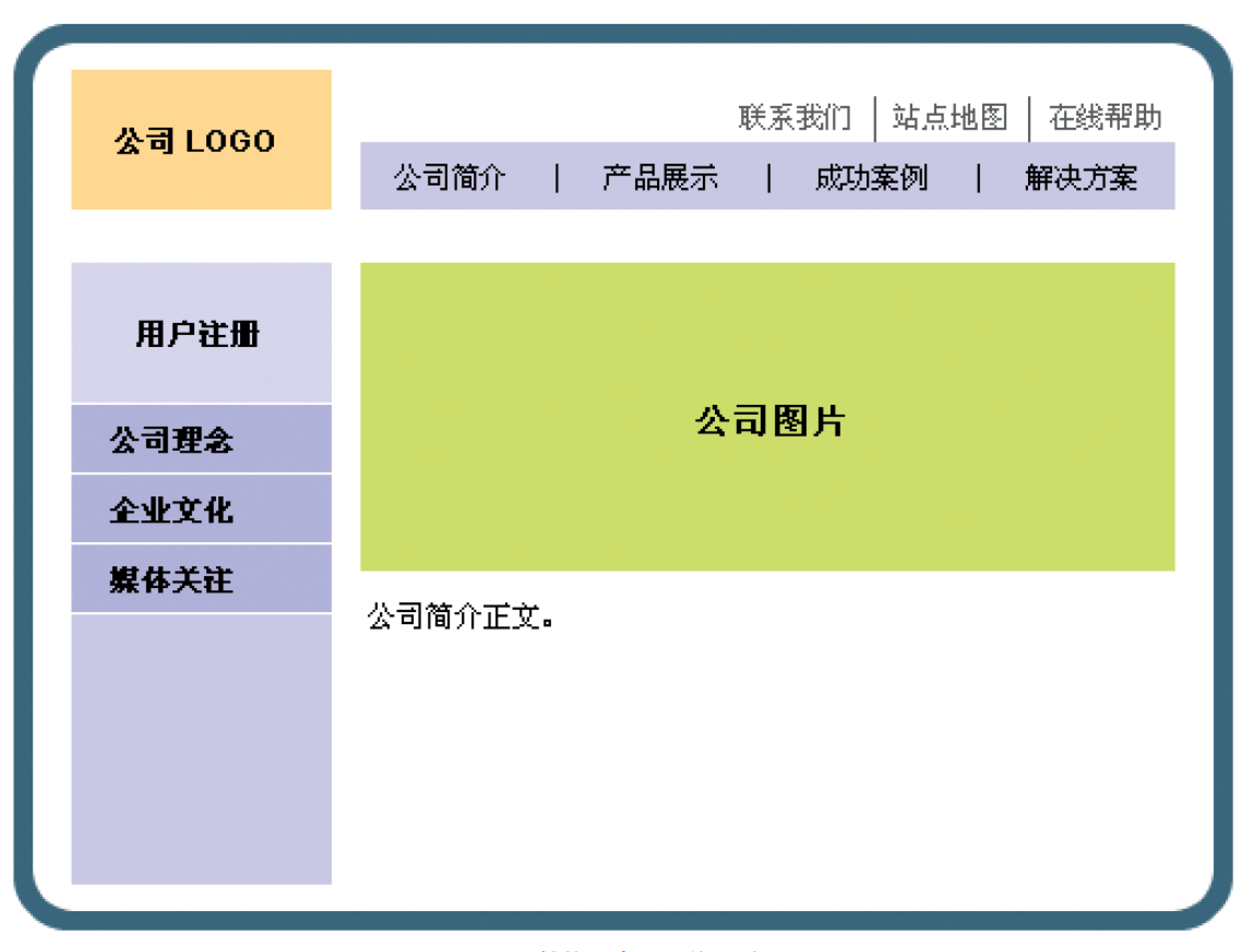
那他们在这种情况下会怎么做呢？他们会绘制出平面图，并且用三维软件制作出最终的效果，可能还包含多个视角。这一切都是为了让你能够一目了然地理解他们的设计意图，并且看到其模拟出来的效果。通过这一方式你就能很清楚地了解设计师的构思，也能很明白地告诉他：墙面似乎太花哨、吊灯应该更有工业感等等。而设计师也

不用慌，因为现在还没有开工。他所要做的只是修改构思，再绘制一些效果图就够了，这并不会花费太多时间。

这种装修的效果图就是一种“原型”（prototype）。为了让用户能够有效地评估你的设计，你就应该根据自己的构思来制作一个大概的草案版本，然后展现给用户。

别被“原型”这个词吓倒，它并不复杂，只是设计过程中的一个辅助“道具”而已。它可以是画在纸上的一组屏幕草图，也可以是通过超链接显示的简略图像。原型就是一个候选的设计方案而已，你也不用做得多么完善——只要把你的设计意图充分表现出来就行。

所以，别一想到“原型”就头疼。制作原型并不需要繁琐的过程或者麻烦的编程，它只是一种与用户交流的途径，确认我们对他们的理解是正确的。很多网站设计公司在项目开始的时候都会在策划方案里面绘制一些网页框图，用以向客户表示网页的具体结构。这种用 Word 在几分钟内就能完成的东西其实就是一个原型，只不过很多人没有意识到而已。



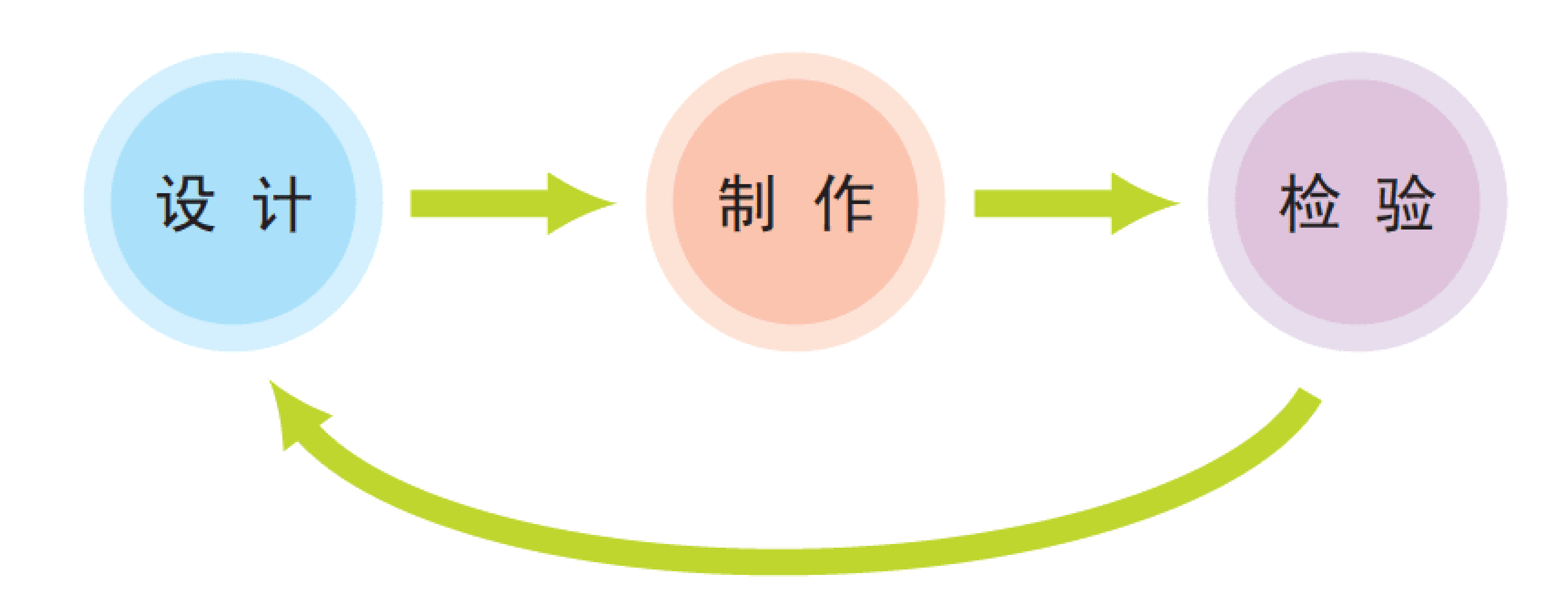
网页结构图也是一种原型

在与用户讨论设计的时候，原型是非常有用的辅助物。而且有了它之后，你与其他开发人员之间的交流也会方便很多，也能够帮助你们选择不同的设计方案。例如你可以指着原型上的某个按钮说：“它应该从这里移到这里，因为刚才那个用户说他没有注意到这个按钮。”

另一方面，如果用户执意希望某种其实并不好的设计，你也可以制作相关的原型来让他们自己尝试一下，看看到底会出现什么样的情况或错误。这样明显更有说服力，而且也表现得更有诚意。

在我们之前提到的“交互设计的 4 个内容”中，原型包括了其中的两个方面：“开发满足需求的一些候选设计方案”和“制作设计方案的原型”。对应的，我也分别向你解释这两个过程。

第一个过程是开发“概念模型”。这一阶段主要关心的是描述产品做什么，如何工作；第二个过程则是物理实体方面的设计，也就是具体的界面设计，关注的是细节，例如屏幕和菜单的结构、图标和图形等等。这也是一个迭代的过程，你将要重复以下步骤：



在此之前我们先看看在工作中可能会遇到哪些原型。

3.2 初级原型和高级原型

根据设计的目的，以及制作上的难易程度，我们可以把原型分为初级原型和高级原型。这种分类也代表了原型模拟程度的高低。所以如果你愿意的话，也可以把初级原型称为低保真原型，把高级原型称为高保真原型。

3.2.1 初级原型很重要——初级原型的多种形式

初级原型，或者说低保真原型，指的是与最终产品不太相似的原型，甚至根本就无法使用。对于工业设计来说，初级原型连使用的材料都不一定需要与最终产品相同。

在设计 Palm Pilot 掌上电脑的时候，设计师用木头雕刻了一个与设想的形状、大小相仿的模型。他把这个模型随身携带，在各种地方“假装”输入信息，从而体验拥有这么一种设备时是个什

么感觉。

这个木头玩意就是一种初级原型。非常简单，甚至有些奇怪。但是它能够模拟使用场景。之前你看到的网页结构框图也是一样，它与最终的页面之间还有很大的距离，用户不能仅靠这个框图就能使用网站。

但是，用户能够凭借这些简单、粗略、然而却又是明显的原型来产生最直观的概念。比如说某些人可能觉得那个木头掌上电脑拿在手里像块砖头，那么设计师就会去考虑是否可以把产品的外形尺寸再缩小一些。

在前面那个简略的网页结构框图中，也许用户会感觉在主窗口放置那么大一副公司图片似乎有点夸张，那么他们就能马上提出来。而如果你没有给用户图例，仅仅只是说“我会在主窗口放一副大尺寸的公司照片，这样可以增加感染力”，那么用户就不会对这个所谓的“大尺寸”有什么直观的感觉。没有直观感觉就不会提出意见，那么在最终网页设计完成之后用户再提意见的时候，美术设计师和页面制作人员就得花费一定精力去修改了。

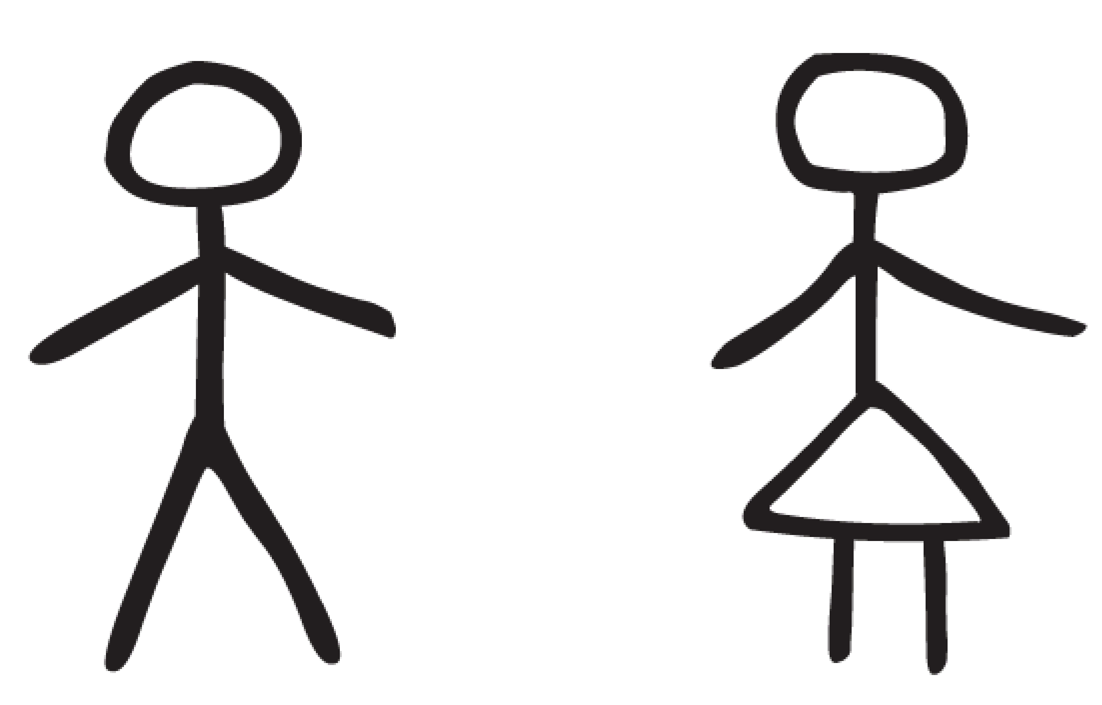
初级原型的优点是简单、便宜、易于制作，这就意味着它易于修改，从而适用于我们尝试不同的设计方案。而且正因为它没有修饰，所以用户能够把注意力集中在你所展示的部分。在产品开发的初期，这种灵活的原型非常适合于各种探索、尝试和修改。

初级原型有很多形式，任何人都能够制作它们：

草图，或者说涂鸦

初级原型可能会用到很多草图。对美术设计师来说这当然是“洒洒水”而已，但如果是程序人员或者项目经理，他们拿着笔的时候也许就会有些挠头。但这只是草图而已，不是艺术品，哪怕是小孩的涂鸦，只要他们画出了特征，我们就能看懂。所以，千万别对自己的绘制能力失去信心。

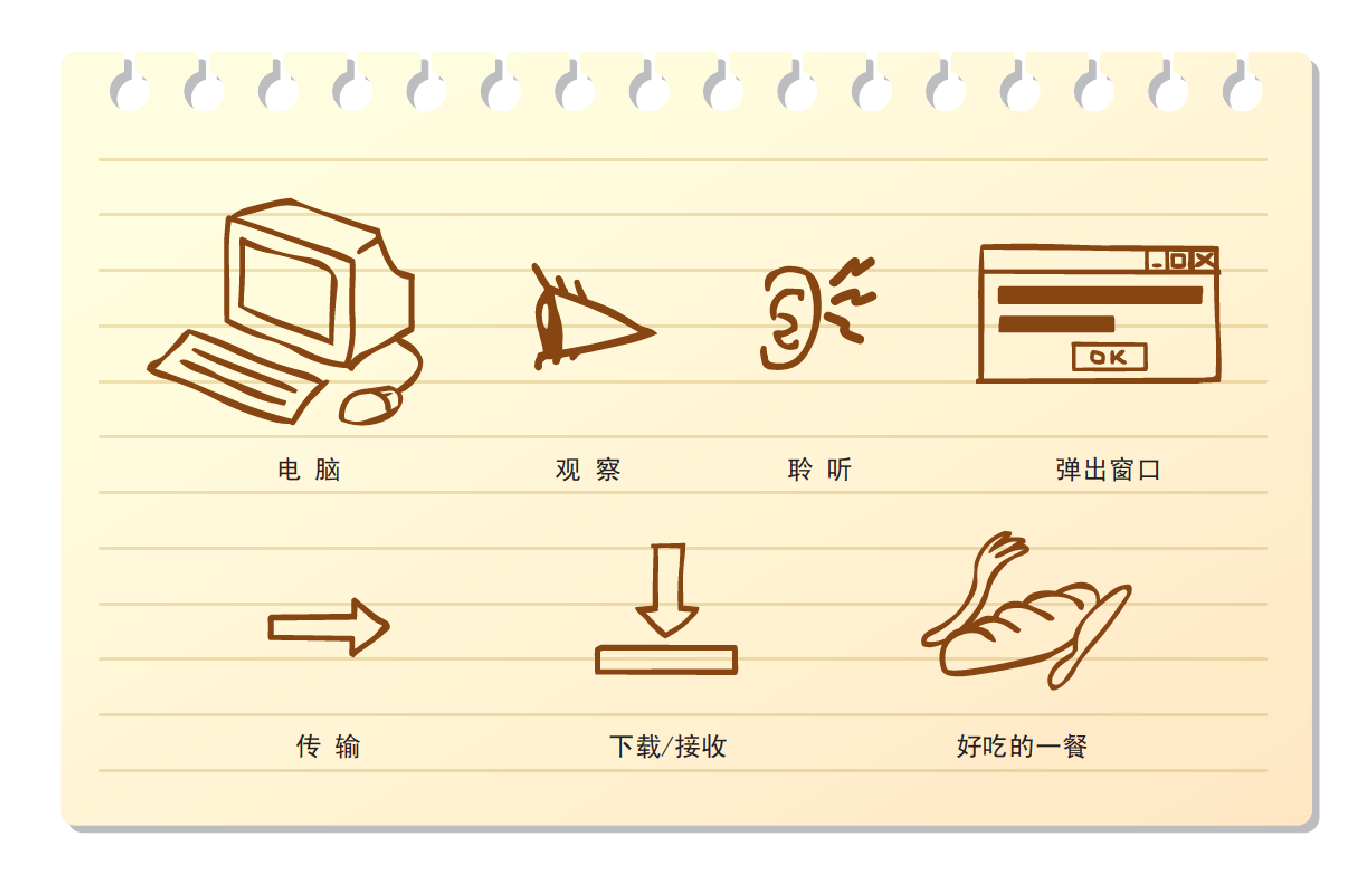
一般而言草图都是非常简略的，类似符号和图标，而且更为抽象。任何人都能画出草图，例如下面这个：



男性用户和女性用户

嗯，我觉得这个画得还蛮不错的。别跟我说你连这种草图都画不出来。

你有时候可能需要绘制一些人、物（例如计算机、书本、打印机）和动作（例如发送、查找和传输数据）。如果是绘制界面草图，你还需要画出一些图标或者对话框等等。和前面一样，你不需要精雕细琢。以下是一些例子，花个几分钟就能做得很好。



连环画

这种原型更专业的称谓是“情节串连图”。情节串连图通常与场景（也就是“讲故事”）结合使用。

所谓情节串连图，就是画出一系列的草图（连环画），说明用户使用产品来执行任务的过程。这种连环画需要包括具体的人物和环境。

之所以要绘制情节串连图，是因为它能够补充更多的细节（记得“一幅图顶得上1024个字”这句话吧）。看着图，用户就能通过一种“角色扮演”的方式把自身代入到场景里面，逐步体验场景和交互的过程。这种更为具体的方式能够让你和用户都有更大的几率发现问题。

例如下面这个情节串连图，它描述了我之前举的一个例子——用户去某家书店购买图书。



1. 周日去书店买一本书
2. 书架上陈列的太多找不到
3. 使用查询终端的人很多
4. 提示需要用户名和口令

很明显图没有画完。但从这几幅（可以说很简陋的）图里面我们已经很清楚地看到了一些问题：

- 由于周末顾客量较大，用户在尝试使用查询系统的时候很可能得排队，也就是要被迫等待一段时间。这时候他们肯定不会很愉快。
- 不太愉快的原因同时也可能在于之前在陈列架上寻找图书未果，用户被迫使用查询系统。
- 用户名和口令在此处显得非常突兀。为什么需要用户名和口令？
- ……

我认为这个难度也不大，不是吗？

制作卡片

在开发 Web 或 Web-based 产品的时候，最常用的办法是制作卡片。

制作卡片，就是把纸张裁剪成一些 A5 大小的卡片（当然你可以随意指定卡片的大小），然后把已经初步设计好的界面草图绘制在卡片上。一张卡片代表一个屏幕截图，或者任务的一个步骤。你需要把这些卡片依次展示给用户，根据他们的不同选择（也就是虚拟的“操作”）给他们不同的卡片，表示不同的结果。

这就有点真正“交互”的意思了：用户进行“操作”然后得到相应的结果。在后面我会给出

一个具体的例子，说明如何使用这类原型。

模拟界面

模拟界面还有个名称叫做“Wizard of Oz（奥兹国的巫师，来自《绿野仙踪》）”系统。它其实就是“制作卡片”，只不过这次的卡片不是纸做的，而是在计算机上显示。

当然，这些界面同样也不是真正的产品，它们都是事先“画”出来的。在这里面也没有真正的交互。用户坐在计算机前“操作”这个原型，就好像看到的是真正的产品界面，但实际上这台计算机被连接到另一台计算机上。在那边有一个操作员通过用户的举动来选择需要展现的界面，然后在用户的机器上显示。这种响应其实都是人为的。

这其实就跟卡片模拟一样，只不过你没有说“好，稍等，我找找现在应该给你看的卡片”。这一切都隐藏在那台机器上面，用户并不知道下一个界面出现的过程。



模拟界面的好处是，直接面对计算机“操作”比卡片更为真实，有助于用户更顺利地“代人”自己的情感，就好像确实在使用一样。

通过上述的这些初级原型，你就能判断用户使用时的心理，而且可以验证界面元素是否合适、功能是否充分。但是，它们毕竟是初级的、低保真度的原型，你不能通过它们体现出用户使用时的速度、效率（例如软件的响应时间）。其他一些方面的特征也无法体现，例如声音和美术设计等等。

不过，初级原型最大的优势就是简单方便，你能快速地获得反馈意见。所以它们仍然有很高的价值。

3.2.2 高级原型同样重要

相对于初级原型来说，高级原型与最终的产品更为接近。如果说木头方块是掌上电脑的初级

原型，那么它的高级原型就是带有按键的高精度模型。

在你去逛手机店的时候，售货员小姐不会直接就把真机给你看。她会递给你一个手机的精模，外观几乎和真机一模一样。很明显，这么一个精致的模型肯定比木头块要好得多。这就是用户体验高级原型的方式——除了功能之外，其他一切都模拟得惟妙惟肖。

既然需要做到惟妙惟肖，那么仍然使用纸张或卡片就显然达不到要求了。你需要用到一些工具。常用的原型制作工具有Director（现在应当归于Adobe公司旗下了）、Visual Basic和SmallTalk等等。这些软件拥有很完善的开发环境，功能强大，制作起来也能得心应手。你甚至可以只需简单地拖放图形元素就能组建图形界面，而不必编写任何代码。

高级原型其实也可以看作是“Wizard of Oz”的智能版。当用户进行了某个操作之后，原型能够自我响应，而不再需要有人在背后操纵。其实系统仍然不是真的在工作，这一切也还是模拟出来的效果——它其实还是一个木偶，只不过我们事先花了一些精力使它能够进行自我控制。

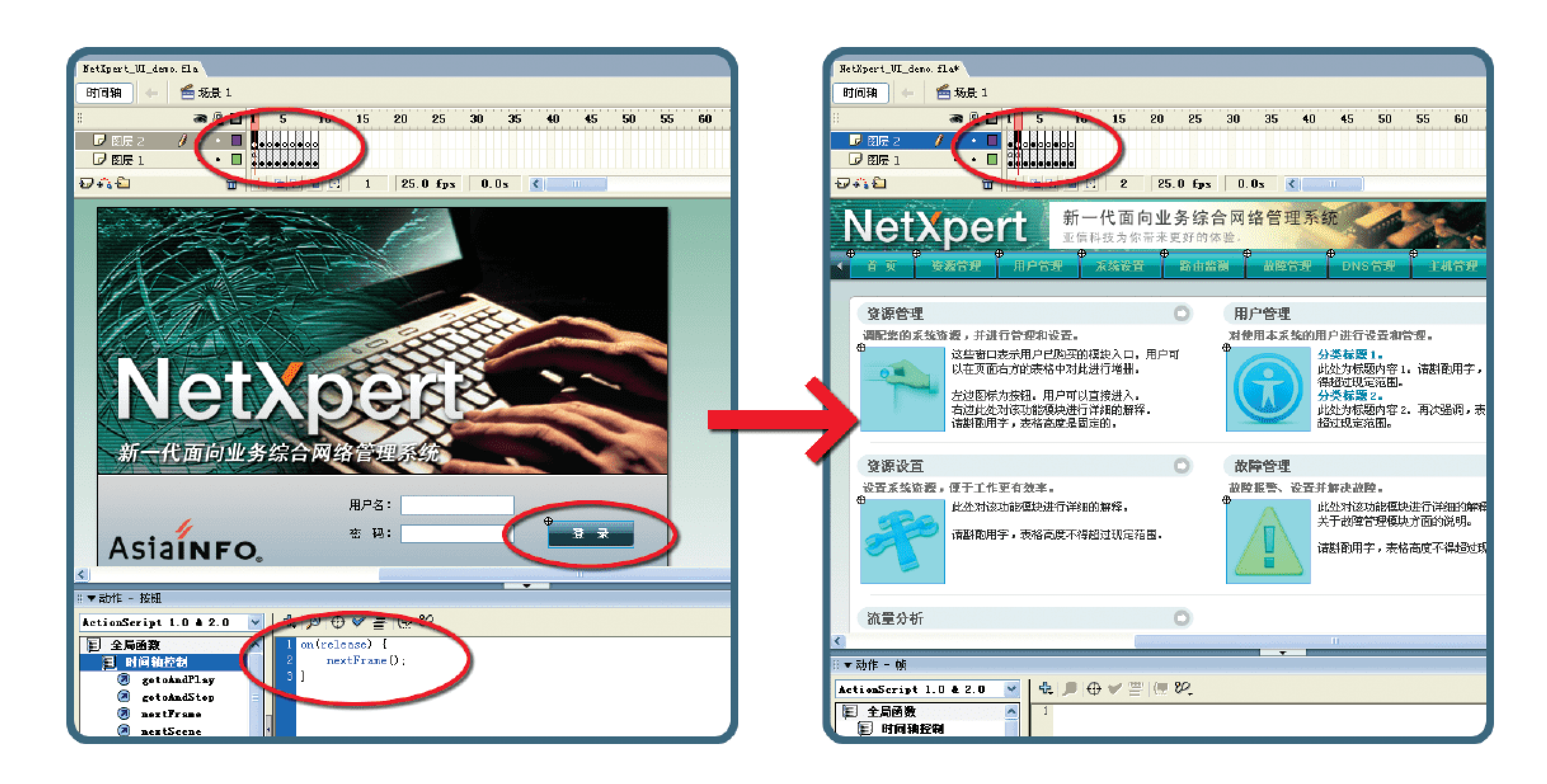
通过这些软件制作高级原型，其实也就是通过软件告诉这些界面，在用户的什么操作后需要出现什么结果。这些都是事先制定好的。但在用户的眼睛里，它基本上就是最终的产品。

对于Web或Web-based产品来说，制作高级原型则更为方便。因为它们本身都是通过Web浏览器来使用，使用超文本链接来连接各种界面和元素。甚至可以说，制作高级原型和真正的网页产品使用的工具都是一样的，例如Dreamweaver。你只需有一点网页制作的基本功就能够制作高级原型，所用到的元素也无非就是表格、超链接和按钮之类。当然了解一点JavaScript会更加合适。

此外，Macromedia的Flash也是一个强有力的工具。软件提供了非常强大的图形设计功能，而且自身就提供很多界面控件，根本不用你去制作。最重要的是，它的脚本控制能力对于简单的模拟界面交互简直易如反掌。

在Flash中，你只需要把每个可能显示的界面绘制出来，然后逐一放到每一帧中去。你可以用很简单的脚本控制界面中的按钮，告诉它们当用户点击时应该跳到哪一帧。那一帧所显示的就是应该出现的界面。你还能够加入声音、动画等其他界面元素。

例如下面的Flash模拟界面。在第一帧我放了一个系统的登录界面，而第二帧则是该系统的首页。在第一帧里面，“登录”按钮的脚本告诉它“当用户点击后，场景跳到后面一帧”。就这么简单。



虽说高级原型有种种的优势，但也有一些专家认为它仍然存在很多问题。比如说：

- **制作时间长**

高级的东西总是要花费大把的时间。

- **评估的时候更容易专注于表面问题，而不是产品的本质**

高级原型做得太“以假乱真”的话，用户可能会过多注意那些图形、美术和表现力等方面。但是你的目的是为了让用户判断产品是否满足其使用的需要。

- **开发人员不愿意修改花了很长时间才制作出的东西**

很自然，我花了十天时间和无数心血做了这么精美的一个玩意，你们说不好用就给否决了？然后我要再花十天？

- **高级原型有可能会把目标定得过高**

一切愿望都是美好的。制作原型的时候不需要考虑真正的实现问题，所以原型越精致，就越会投入精力，同时也就更加容易加入自己的想法，哪怕是天马行空。但最后很可能实现不了。

- **高级原型中的一个错误就可能打断整个评估过程**

精密的东西总是更有可能出错，越高级的东西很明显也越难以修改。如果在评估过程中发现原型出了毛病，你很可能要对用户说：“真对不起，我需要一到两天的时间修改它，所以今天就到这里吧。至于你们到这里来路上所花的两个小时……”

在这里我列出了一张表，概括了初级和高级原型的优缺点：

	优点	缺点
初级原型	开发成本低 方便快捷，随时能体现设计概念 便于和用户交流 能获得快速的反馈意见 适用于搜集用户的需求	可发现的错误不多 不能作为具体的设计规范 建立需求后，作用有限 可能会漏掉很多可用性方面的问题 不能完整地展示任务流程 受到制作物件的影响
高级原型	包含完整的功能 完全可交互 用户可以自主操作 体现了明确的任务流程 用户能获得最终产品的使用体验 适用于探索和测试	开发成本很高 制作耗时 有时候可能很难修改 不太适于搜集需求

注意，列这张表的目的不是在告诉你哪种原型更好。初级原型代替不了高级原型，反之也一样。在探索产品的内容和结构时（或者在刚开始面对某个问题时），你应该使用简便、便宜的初级原型；但当问题已经研究到了某个阶段，你需要考虑细节，或者向其他人讲解设计的时候，仍然得使用高级原型（即使会比较麻烦）。

3.2.3 嗯，不过原型仅仅只是原型

记住，设计和制作原型只是正式开发工作之前的一个阶段。这个阶段的目的，是为了"快速制作某个东西，从而测试产品的某些方面"。它是一个权衡折衷的产物。正因为初级和高级原型都存在着这样那样的缺点，所以它们都无法代替最终的设计成品，这是毋庸置疑的。

原型只能帮你搜集到更多信息，这些信息用于改进原型。当原型基本上能够满足所有的情况时，你才能进行正式的开发工作。

在制作原型的时候，你也可以从另一个角度对它们进行分类，分别是"水平原型"和"垂直原型"。

- **水平原型**展示了产品的绝大部分功能，但都不详细。这是一种横向的角度。在对用户展示整个产品的设计概念时，你需要让他们了解产品的每一个部分，它们的结构以及相互之间的关系。这种时候水平原型最合适不过，因为这个时候没有必要对某一部分进行详细分析，你们关注的是整个层面的问题。

- 与之相反，**垂直原型**只展示一小部分功能，但足够详细。它是一种纵深的概念。在用户开始把注意力放在某些详细功能上面的时候，你就需要把垂直原型展示给他们，然后一起探讨相关的细节问题。

从这个角度来说，不存在一个既“水平”又“垂直”的全面原型。因为一旦你把所有设计都集中在这个原型里面，它就变成一个“成品”了，那样等于说你根本就没有制作原型这一过程。你会耗费大量的时间来制作它，直到时间和各种资源以及你的感情都不允许你把它抛弃。

也就是说，你会不由自主地对它进行反复的修改，而不是仅仅把它作为一个参考。当出现了问题的时候，你会疲于弥补之前的错误。这会浪费很多时间，最终的成果也很难令人满意。这一点我们在本章第一节就讨论过。

你可能会这么想：“也许我能从原型开始，到最后把它直接变成最终的产品。”或者：“把几个垂直原型拼到一块，不就是成品了吗？”

有些公司确实是这么做的，但我认为并不合适。因为原型本身是一种折衷的产物，并不是产品的雏形。它只是一个“外壳”，而里面是空的。尽管原型也需要让用户来评估，但评估的主要精力是放在如何完成任务方面，并不是产品的质量。

而且在制作原型的过程中，你可能经常都会做一些让步。这些让步不是很明显，甚至你自己都会忘掉（甚至意识不到）做过哪些让步。例如下面这些隐藏的问题：

- **原型内部结构的设计可能不够完善**

  你可能只考虑了这个原型所包含的几个功能，但没有考虑它们能与其他功能之间的关系。

- **有可能包含一些错综复杂的代码**

  你在制作原型的时候可能会想：“毕竟这只是个原型，这些小毛病等我到后面正式开发的时候肯定会修正过来。”但说不定第二天吃早餐的时候你就忘了。

- **功能划分不一定合理**

  不同的开发人员在制作原型的时候可能只关注于自己那一部分。如果把不同开发者制作的原型拼凑到一起，很可能会出现重复甚至排斥的情况。

- ……

如果在项目开始的时候整个部门就仔细地规划，进行了详细的预先定义，那么你也许可以把原型逐步演变为最终产品。但是这个方法会使测试和维护问题成为日后的隐患。可能最终你仍会发现，产品的质量还是下降了。

不过，有时候你也需要“抢滩登陆”。如果你开发的是一个创新的设备，那么尽早推出一个“足够好”的产品来占领市场，也要比晚竞争对手两个月上市好得多（即使你那时推出的质量更好）。因为它的价值更多在于“新”。这也是很多软件 Demo 版就面市的原因之一。

3.3 概念设计，或者说初步设计

之前我们谈到了很多原型的设计和制作方法。那么，在哪个阶段使用这些原型？如何使用它们？现在我们就来讨论一下这些问题。

不过让我先澄清一个事实。

3.3.1 伙计们，让我们先考虑功能

当要开发一个 Web 或 Web-based 产品的时候，很多设计人员总是马上开始试图确定它看上去会怎样。他们中间有的用纸笔或 Photoshop 勾画设计；有的用 Dreamweaver 安排产品的外观和控件；还有些人直接就开始编写实际的实现代码。

但是，一开始就去担心外观是本末倒置的。那样很诱人，但事实证明那常常是一个错误。屈服于这种诱惑会导致产品可能"看上去很美"，但却缺乏必要的功能，里面还可能有许多组件和函数与用户要完成的工作毫不相干。而且这种产品对用户来说感觉好像是一时兴起的产物，因此很难学习和使用。

所以你应当首先考虑功能，然后才是表示。

但别误会！ 这句话并不是说你要"首先设计并实现功能，然后再担心用户界面"。很多程序设计人员和项目管理人员一直都秉持这种看法，也就是功能比界面更重要。这同样是错误的。靠这种想法无法制作出成功的软件。用户界面可不是在项目开发结束的时候才需要"担心"或添加的什么东西。我们整本书就在讨论用户界面设计，你认为呢？

"首先考虑功能，然后才是表示"这句话的正确意思是，你应当在考虑用户界面的外观之前先考虑产品的目标、结构和功能。在这里并不是说要你先实现出功能，而是要搞清楚自己到底应该做什么。

在你勾画外观、布置控件或编写代码之前，应该先集中精力回答如下几个问题：



你需要尽早回答以上的问题，而不是在正式设计中才去修改。同时你还得考虑如何让用户去理解产品。这需要为产品开发一个“概念模型”。

3.3.2 概念模型是个什么玩意？——设计概念模型时要遵循的原则

概念模型可不是个“玩意”。它非常重要。简单来说，概念模型就是一种用户能理解的模型，它能够描述系统应该做什么、如何运作、外观如何。看了前面关于“原型”的描述你应该明白，概念模型正是靠原型来体现的。

但是，我们之所以叫它“概念模型”，就是突出“概念”的重要性，也就是表现出你对产品的设计概念。它应该能够抽象地说明产品的功能，以及展现给用户的方式。根据概念模型设计出的用户界面将是最为整洁、简单的，也更加容易理解。

当然，在设计用户界面之前去开发一个概念模型通常都很困难。对我们这些开发人员来说，直接讨论诸如控制面板、菜单和显示这样的界面元素会更加吸引人。另外，有些销售人员也需要你的最终界面来招揽客户。他们很可能会说 ：“差不多就行了，赶快做漂亮点，我马上要拿这个设计方案去见客户，而不是在这里等你们无穷无尽地讨论来讨论去。”

遇到这种情况，你应该温文尔雅地拒绝他们。没有正确的概念模型，就不会有成功的用户界面。这一切本身就是从用户的利益角度出发的。

设计概念模型的时候，你需要遵循以下一些指导性的原则 ：

开放思路，同时考虑用户和应用环境

还记得环境吧？之前我们说过，我们必须考虑在任务过程中用户所处的环境，也就是产品的应用环境。同时，你需要把所有涉及到的用户都考虑进来，不要忽略任何人！

保持简单，但也不要过于简单

最好的办法是使用最少的概念来提供所需的功能。与此同时，别让那些任务中并不出现的概念（例如数据库、字符串和缓冲区）去搞晕用户。另一方面，也不能在降低复杂度的同时把一些重要概念给省略了，或者是把用户们认为本来是分离的概念给合并了。这和包含不相干的概念所导致的结果一样，用户同样会被你弄晕。

使用初级原型来快速获取反馈

别在这个地方花费太多时间。没错，做一个精美的模型非常有成就感，但是这不值得。绝大多数模型都需要修改。而且记住这仅仅只是模型，还没到正式设计的时候，让我们把这方面的精力集中在有一定把握的时候会比较好。

反复迭代进行设计

这个已经反复强调过了。肯定会有很多问题在现阶段没有考虑到，你需要不停地根据问题反馈进行修改和再设计，直到大家都认为差不多了为止。记住，“要获得一个好想法，需要先获得许多想法”。

3.3.3　开发概念模型的几个问题

要开发概念模型，你就需要根据用户的需要和需求来规划这个产品。以下几个问题是非常重要的 ：

采用什么样的交互方式?

交互方式就是指用户与系统交互时的活动类型。常见的交互方式包括 ：

- **指示**

用户向系统发出指令，例如输入 DOS 命令、选择下拉框列表、点击按钮或超链接等等。

- **对话**

用户与系统对话、提出问题，例如你使用搜索引擎的时候，需要输入关键字。

- **操纵和导航**

用户利用在现实世界中积累的知识来操作对象，例如拖曳某个窗口，或者用电脑玩 FIFA。

- **探索和浏览**

用户发掘和浏览信息，例如去新浪或者搜狐。

- ……

这看起来似乎很简单。对于你的设计来说，哪种类型的交互方式是最合适的？这取决于用户在使用产品时所进行的活动，是在搜集用户需求的阶段得出的结论。同时你还需要知道两点：第一，这也取决于你开发的系统类型是什么 ；第二，大多数系统都需要结合不同的交互方式。

Google 提供的“文件”服务是一个在线编辑并储存、共享文档的 Web-based 产品。其中的“新建文档”、“删除文档”等操作就适合采用指示性交互方式，用户可以直接点击“新建→文档”来新建一个文档文件进行编辑。这种时候就不必使用对话框来显示。

而用户在搜索需要的文档时，显然对话型交互就更为合适。在文本框内输入所需文档的关键字，系统就能回应某些适合搜索条件的文档链接供用户选择。这就如同一个对话过程，用户对系统发出指令，然后系统对其进行答复。

而在搜索结果页面，用户又以探索和浏览的方式来通过超链接进行信息的浏览。



用户点击左上角的“新建”菜单进行“指示性交互”

是更“智能”还是更“服帖”？

概念模型的基本任务是找出产品所要支持的任务。但是在这些任务中，哪些部分需要用户承担，哪些又该系统承担呢？

例如一个进行在线日程安排的产品，系统可以根据用户输入的条件提出满足要求的一组时间，便于用户安排日程。但你也可以不这么做，也许可以让系统自动确定时间，然后向用户发送电子邮件请求确认；或者完全由用户自己输入具体的时间进行安排，系统只是单纯地显示日程表。

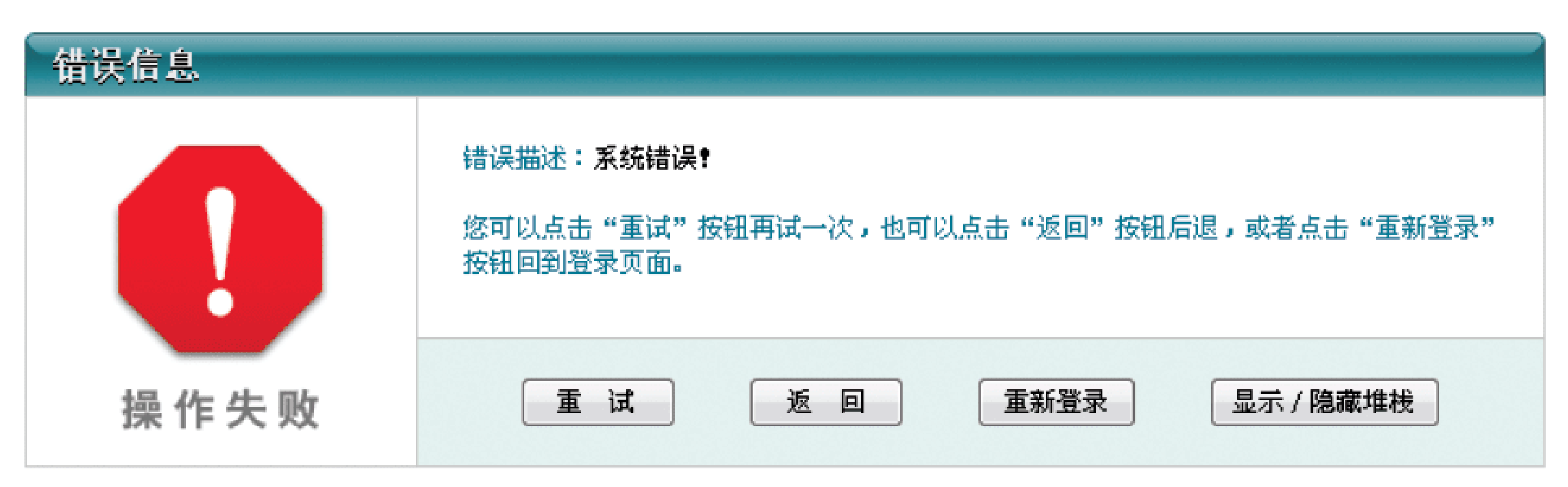
很明显，能自动确定时间的系统是更“智能”的系统，而需要用户自己输入具体时间的系统则更为“服帖”，完全根据用户的要求行事。“服帖”的系统需要用户更多的操作，比较麻烦；而“智能”的系统则可能不够灵活，这种“强制性”也可能引起用户的反感，根本就不会去用它。

系统承担哪些任务，或者承担多少，这都取决于你。但是用户是否接受它，你就无法控制了。所以，多做一些原型进行评估，得到的结论就能更“经得起考验”。

是否存在合适的熟悉概念进行映射比拟？

“映射比拟”是一种把概念与某个物理实体相联系，提炼其相似性的界面设计方式或技巧。例如Windows的“回收站”，就使用了现实世界中垃圾箱的概念，每个人都能在第一次就理解它，

而且很难忘记。同样，Windows 的“文件夹”，也就是现实世界中的文件夹，你可以把零散的文件放到里面，还能随时整理排序。此外还有很多相关的例子，例如弹出的警示窗口，使用的图标是根据交通警告标志演化而来的。



红色的图标比拟的是“禁止通行”的交通标志



“锁”表示用户无法继续操作

这种映射比拟的技巧相当成功。因为它提供给用户一个非常熟悉的参考，有助于学习和理解。不管你是外行还是老手，都能很快地熟悉它，顺利地理解界面功能。

在进行映射比拟之前，你必须仔细权衡什么样的比拟才是最恰当的。同样这也需要你正确地理解用户和应用环境。

如果说现在开发的是一个供儿童使用的数学基础教育软件，可能相当一部分人会选用“教室”进行映射比拟，也就是一个老师站在黑板前，还有讲台和粉笔。但是如果你仔细研究了用户（也就是那些注意力容易分散的小孩）之后，就不再会觉得老师和黑板是个不错的主意了，因为这些根本无法吸引他们的注意力。而且说不定他们对黑板和老师还很反感。所以你应该选择一些儿童喜欢的东西，例如马戏团或者游戏室来映射。

所以，你必须了解系统要怎样才能最恰当地适应用户，找到那个切入点。

当然，也有些“比拟”设计不太成功。例如 Mac OS 的“垃圾箱”。我在第一次使用 Apple 电脑时完全不知道该如何才能取出光盘。后来才知道是把光盘图标拖到垃圾箱上面。我们都知道要

想删除某个文件应该把它拖到垃圾箱上，但弹出光盘为什么也是这样的操作？我相信世界上有无数个人与我有相同的疑问 ："等一下，这样会清除它吗？"



3.3.4　在概念设计中使用原型

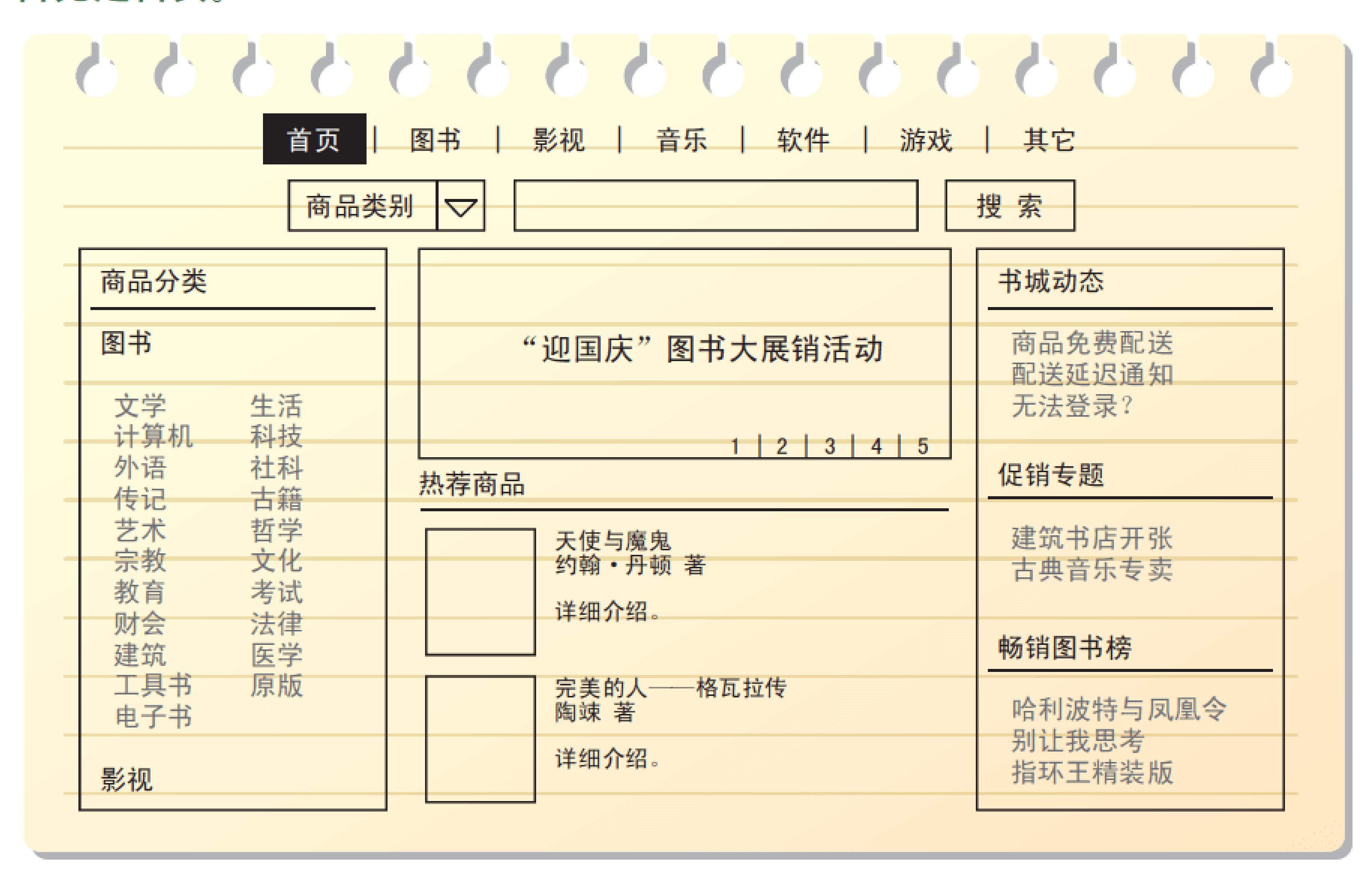
在概念设计中你需要通过制作并展示原型来对设计构思进行检测。也就是说，你要把设计的概念表现成原型，然后展示给用户，听取他们关于这种设计的意见。

之前我们说过，开发的不同阶段需要使用不同类型的原型。一般来说，初级原型（例如草图或者卡片）适用于刚开始设计的阶段，而高级原型（例如可运行的软件）则适用于设计中后期。

原型的作用是要回答某些问题。好的原型应该能够让用户体会到产品将如何支持他们的任务。

一个在线书店的网站，最合适的方法是制作一系列初级的卡片原型。我把初步设计好的页面结构绘制在若干张卡片上，每张卡片代表一个步骤中计算机显示的屏幕截图。这些卡片将被依次展示给用户，根据他们的不同"操作"会有不同的卡片表示不同的结果。

首先是首页。

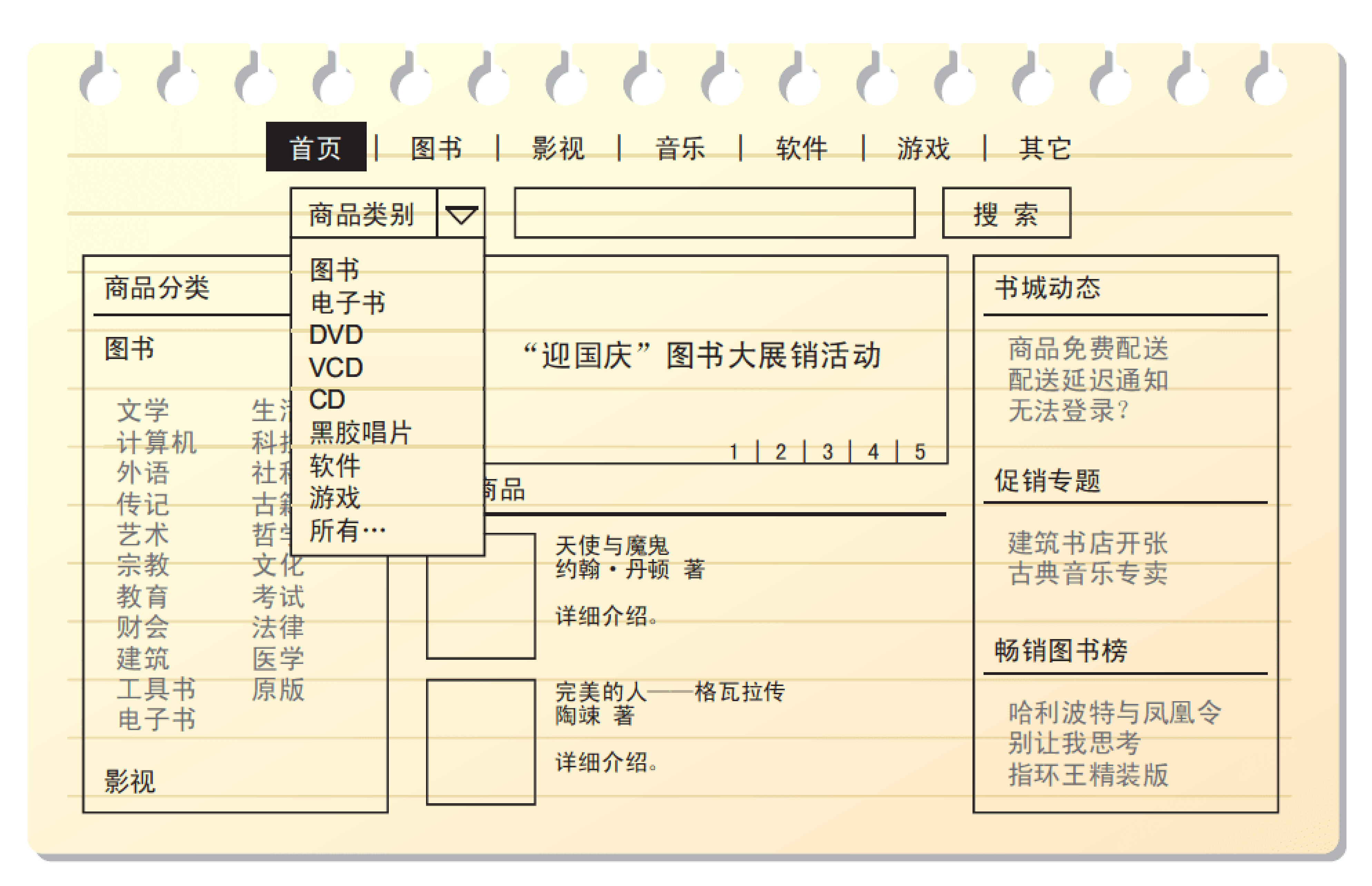


在线书店的首页

卡片虽然很简略，但清楚地描绘了网页的大部分元素，例如顶端的导航栏和搜索工具、左侧的商品详细分类、右侧的个性化栏目等等。同时我们也适当地加入了一些“虚构”的内容，例如中间“热荐商品”里面具体的小说名。这样明显要比“图书 1”、“作者 1”的代号好得多。

当用户看到首页之后，没有提出什么问题。接下来他会根据自己的目标来尝试各种操作。也许他没有明确的目的，只是想看看文学方面有些什么书，然后再决定是否购买，那么他就会在左侧选择“文学”的超链接。此时就需要向他展示“图书”里面“文学”栏目的首页面卡片。或者他有明确的目的，例如想购买一张《变形金刚》的 DVD，那么可能就会直接选择搜索。

这一次的用户节奏比较快，他告诉我希望直接搜索，而且会选择商品类别。那么这个时候我给他另一张带有商品类别下拉框的卡片：



搜索时的卡片

这个时候用户可以根据下拉菜单的选项进行选择。但他产生了疑问：“为什么下拉列表里面的分类和导航栏不一样？为什么是‘VCD 和黑胶唱片’？如果我想买‘磁带’怎么办？嗯，这个有点莫名其妙，我不喜欢这样。”这个时候我就得把这个问题记下来。也许之前对用户需求的认识出现了偏差，也很可能因此需要对原型加以改进。

把问题记下来后，用户会继续下去：“好，我知道要选择 DVD，我会在后面的文本框里面敲上‘变形金刚’几个字，然后点后面的‘搜索’按钮。”此时我就要拿出另一张卡片，上面模拟的是“搜索结果”界面。也许他在那个页面又会发现一些问题。

可以看到，这种方式比真正把网页设计完毕后再给用户看要好很多，因为我能够非常方便地修改设计，只需要动动笔就行。而这些问题和信息对于真正的设计是非常有用的。

3.4 物理设计，或者说具体设计

“概念设计”考虑的是产品的设计概念，用于描述系统应该做什么、如何运作、外观大致是个什么样。而“物理设计”考虑的则是更为具体、细致的界面设计问题，例如导航栏的设计、使用什么样的图标、如何组织和规划菜单等等。

有时候你可能很难分清概念设计和物理实体设计之间的分界线。在制作原型的时候，你不可避免地会同时考虑一些细节方面的问题。其实这种担心根本没有必要。不要刻意去区分什么工作应该放在概念设计阶段，什么工作又应该放在物理设计阶段。煎鸡蛋是先放盐还是后放盐，每个人都有自己的习惯。

不过需要注意的是，在概念设计的过程中你应当充分地自由发挥想象力，不能过早地受限于物理设计，否则将可能影响你的创造性，出来的东西死板而毫无灵气。

对于 Web 或 Web-based 产品来说，物理设计涵盖了许多方面，我们也不可能在这里全面地讲述。但是我想可以结合一些通用的界面元素向你介绍一些好的设计原则。

3.4.1 一些具体的设计指南——交互设计的 8 项黄金法则

在设计界面的时候，一定要确保让界面去适应用户。

例如，为了减轻用户的记忆负担，在界面中你可以列出种种选项，并适当地加以图标，而不要强迫用户去记住那些长长的文字选项清单。而为了引起用户对重要信息的关注，你就需要着重突出表现页面中首要的部分，例如使用更大更醒目的字体，或者更特别的颜色，或者在旁边留出更多的空白。

为了确保设计的产品有更高的可用性，人们开发了各种指南、原则和规则。其中有一些很具体，称为“设计规则”；而还有一些则非常抽象，称为“设计原则”，需要在应用之前对其先做解释。

例如，在设计网站的时候，“保持页面简单”是一项非常笼统又非常重要的设计原则，它作用于整个设计过程。但是在应用这条原则之前，我们必须针对用户、任务以及应用环境来详细地解释这项原则。而“控件与窗体的各边距为 7 像素”或者“所有表格每行均以蓝、白背景色相间隔”则是几项非常具体的规则，你只管照做就行了。

我在 1.3.4 节就已经介绍了一些可用性原则，我把它们称为是交互设计的“摩西十诫”（不记

得了就往前翻翻。现在就翻)。它们是由 Jakob Nielson 博士提出的，这位博士是世界公认的可用性权威。

另一个众所周知的原则性设计指南是 Ben Shneiderman（美国马里兰大学人机交互实验室主任。他也是用户界面设计的先驱）在 1998 年提出的“交互设计 8 项黄金法则”。

力求一致性

网站或软件的设计风格需要保持高度的一致性。例如网站的首页需要和每个下级页面都保持一致的风格，软件的菜单都放在屏幕的左上角。具有高度一致性的界面能给人清晰和整体的感觉。

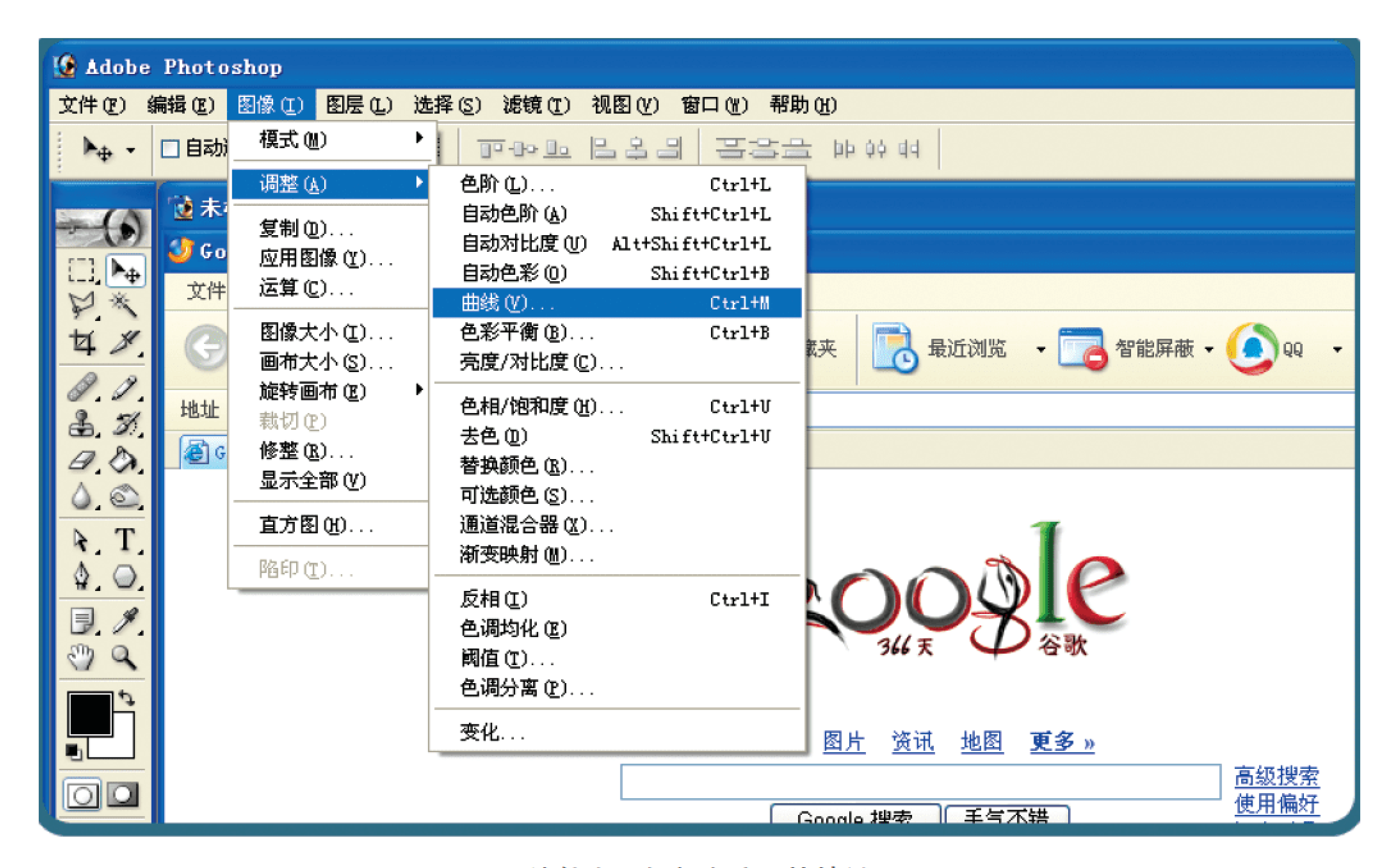


“Google实验室”中一致的logo标识

允许频繁使用快捷键

快捷键表示产品使用的灵活性及有效性。每个人都很懒。想想你每次使用 Google 或百度的时候，是用鼠标点击“搜索”按钮还是直接按“回车”键？

使用过 Adobe Photoshop 软件的人都知道，几乎软件的每一种操作都有相应的快捷键。熟悉这款软件的用户可以不用费事地移动鼠标选择菜单，而只用键盘单击一两个键就能完成指令。例如“Ctrl+M”就等于用鼠标选择菜单中的“图像”—“调整”—“曲线”。有时候看到熟练的设计师运指如飞就好像在弹钢琴，确实很令人羡慕。



软件中一般都会注明快捷键

提供明确的反馈

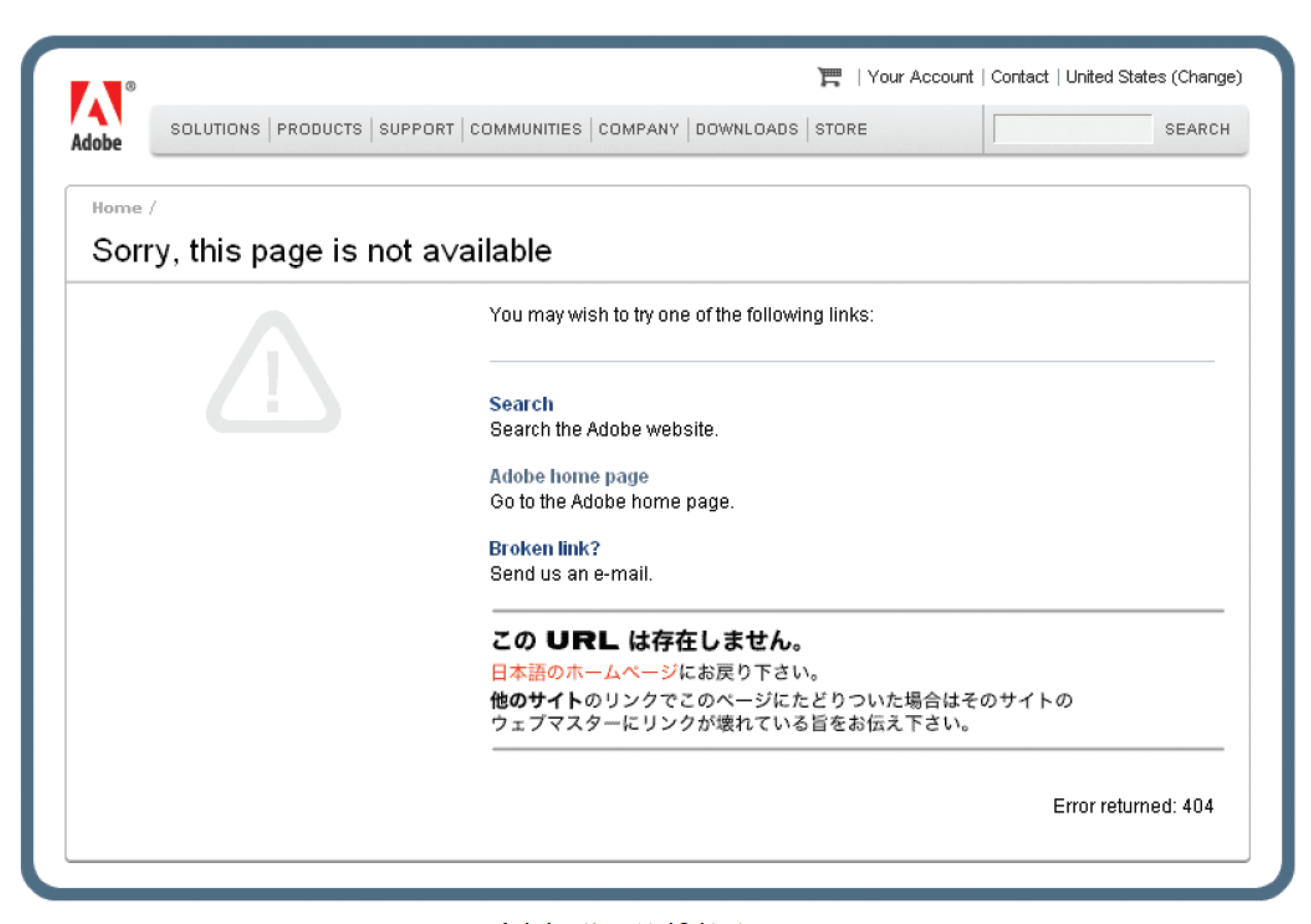
出现错误时，要明确说明错误的含义，而且你需要考虑用户是否能理解。

在几年前还经常出现这种情况：当我们输入某个其实不存在的网址时，Web 浏览器显示“错误 404”。什么是错误 404？也许有些家伙知道“在 http 规则中用 404 来表示某个页面不能访问”，但是我敢肯定绝大多数人不了解这个 http 规则。

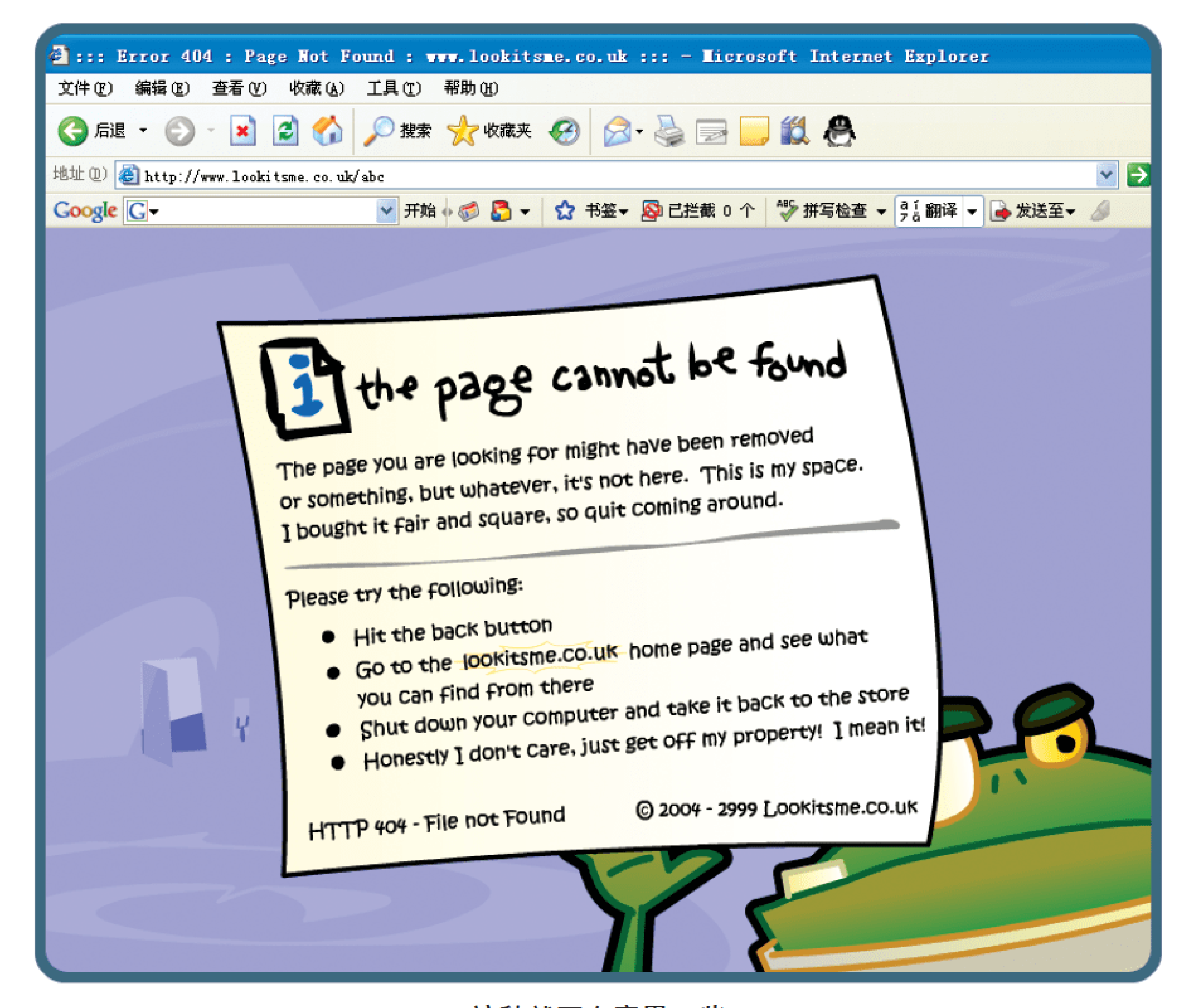
而现在情况已经好多了，当你访问的页面不存在或者打不开的时候，显示出来的提示语起码能让人“看懂”是个什么情况，该怎么办。有些网站甚至自己定制这类出错页面，让用户能得到更友好的体验。



现在的Web浏览器默认404错误页面（比以前好一点）



Adobe公司的404错误页面



这种就更有意思一些

设计对话，告诉用户任务已完成

要在用户完成某项任务或操作后进行提示。如果他们在做了很多操作后却得不到任何反馈，他们就无法知道自己是否达到了目标。那么继续操作的时候他们就会充满疑惑，举棋不定。

这种“提示”并不一定是弹出窗口或者对话框。在网页中很多操作都会根据页面的显示来提示用户。如果用户的每次操作系统都会弹出对话框让他确认，估计也是一种折磨。



黑、黄色的文字提示用户进行的操作已经成功了

提供错误预防和简单的纠错功能

最理想的界面应当能够预防用户犯错误。也就是说，用户在网页或软件中执行任务的时候，你应当提前通过一些办法来避免他们进行误操作。例如可以把某些当前不能点击的按钮设置为灰色。

然而大多数情况下错误总是不可避免的，所以系统应当更为宽容，允许用户在出错后进行纠正。对于一些可能导致数据丢失的操作，在系统正式执行之前应该要求用户确认，允许他们改变主意。

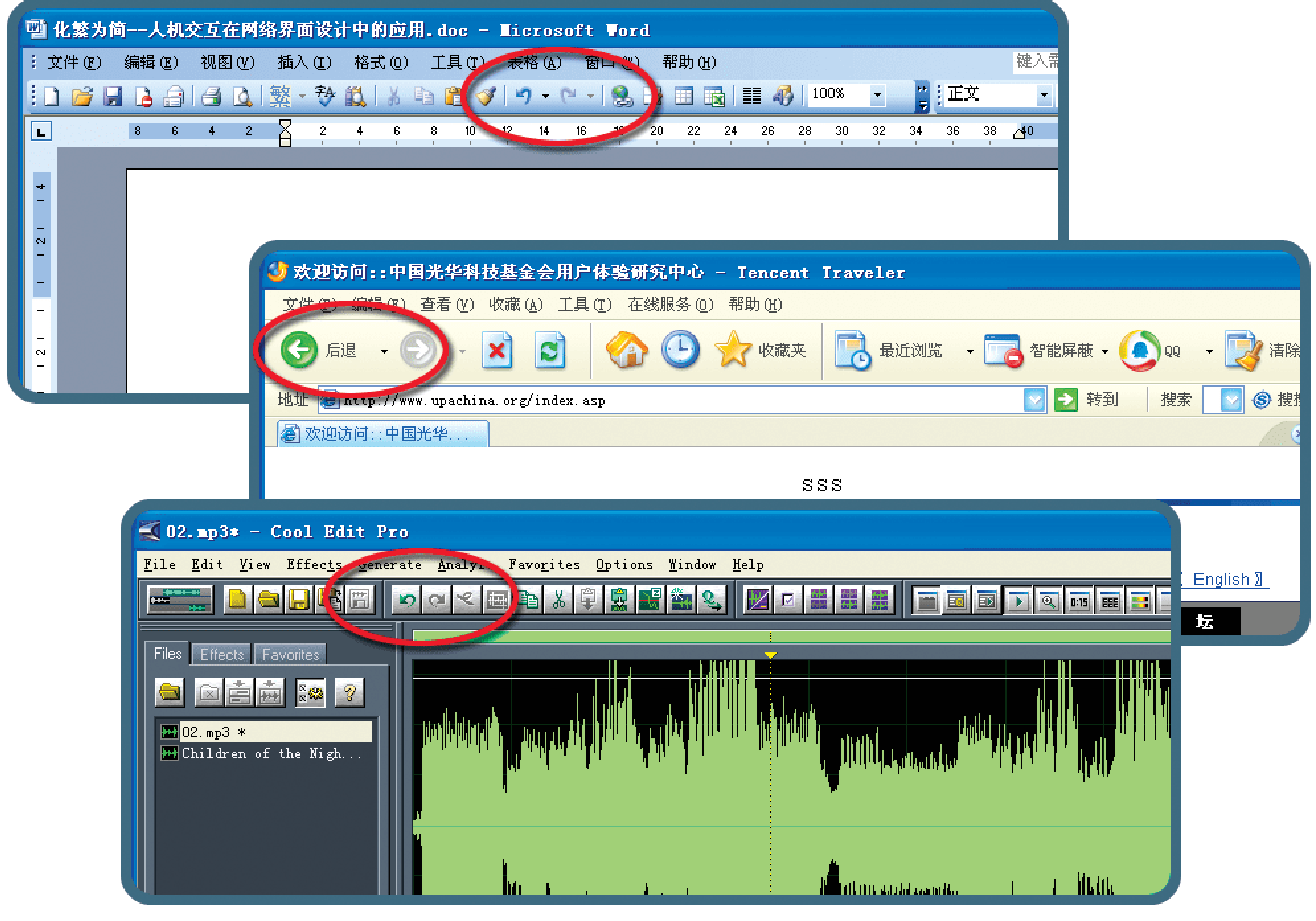


确认一下，别犯错误！

应该方便用户取消某个操作

现在绝大多数应用软件都有“撤销”和“恢复”两项功能，快捷键是“Ctrl+Z”和“Ctrl+Y”，似乎已经成为基于Windows操作系统的业界通用的一项标准。如果用户时刻都因为惧怕“一失足成千古恨”而战战兢兢畏首畏尾，那样的用户体验可想而知。

但是我发现在Windows自带的“画图”软件中，仅仅只提供了三次“撤销操作”的机会。这个限制似乎有些多余，毕竟计算机只有16M内存的时代已经离我们远去了，不是吗？



大多数产品都有多次的“撤销”和“恢复”功能

用户应掌握控制权

一般而言用户都希望能由自己来控制交互，而不是由设备来控制。他们是“主角”、“主控者”，不要让他们感到备受拘束。也就是说，如果用户在任务进程中出现了问题或错误，或者不希望执行该任务，他应该可以随时中止任务或者退出，而不是必须无奈地看着系统继续（甚至是错误地）执行下去。



可以让用户在过程中随时中止下载

很多网站都可能有演示动画或片头，大多数都是 Flash 制作的，而这类动画或视频都需要一定的载入时间。任何人都不应该强制用户在看完演示视频之后才能访问网站其他内容。游戏也是如此，用户可能希望马上开始进行游戏，而不想看冗长的背景故事。在这种情况下，在动画或游戏片头中加一个“Skip”会好很多。



加入“Skip”按钮能让用户直接开始执行任务

减轻用户的记忆负担

我们应该尽可能为用户提供帮助，避免要求他们记住各种信息。例如你可以根据菜单项之间的关系，把它们划分为一些独立的选项组，这样便于用户找到并记住其中的逻辑关系，在以后他们就能很容易地找到所要的选项。

另外你也可以对一些重要功能或选项加以辅助图标，加强用户的视觉感受。人类对于图形的识别能力要高于文字。如果图标做得好的话，用户甚至可以不用看文字，仅凭对图形的记忆力就能很快知道该选择哪个选项。图标的设计应该直观、具有概括性，同时它也能让界面更为美观。



文字选项配以较为形象的图形标识

3.4.2 界面元素和界面风格

当前主流计算机的界面风格被称为“WIMP 风格”，即窗口（Window）、图标（Icon）、菜单（Menu）和指针（Pointer）。

所有的 Web 产品以及绝大部分软件的界面都是由这些元素组成，例如对话框、按钮、文本框等等。我们需要设计的就是这些元素。

有时候一个完整的产品需要特有的风格。这和公司 VIS（Visual Identity System，视觉识别系统）制定的细节相关。例如，有时候一看页面你就能发现访问的是哪个网站，因为它有自己的特色而且给人印象深刻。这包括特定的用色、logo、菜单和图标的设计风格等等。

这其中最具代表性的就是 Apple 公司的网站。尽管他们现在刚刚进行全球性质的改版，但你仍然能轻易地辨认出这就是 Apple。这些细节包括产品的表现形式、图标的材质等等。



Apple公司的新网站

在有些时候，页面或界面的这种风格也可能是通用的，不能太具个性化，例如Windows操作系统或者Office系列软件。因为它们的用户数以百万计，必须容纳各种操作习惯、文化等等的差异和冲突。但同样的，每个Windows产品都应具有相似的形式，即使是最新的Windows Vista，也需要和其他版本有相传承的地方，例如“最大化”、“最小化”和“关闭窗口”的按钮都在窗口的右上角，“开始”菜单在桌面的左下角等等，这都是Windows风格所规定的。一般来说必须遵守。

Apple的Mac OS X操作系统也是一样。它的界面艺术性和普遍性兼备，晶莹透亮的图标设计是其最显著的特征，而且菜单、窗口的设计都有自己独特的风格（注意到“关闭窗口”的按钮在哪儿了吗？）。除此之外，Mac OS X使用了类似动画效果的响应显示技术，带给用户一种舒适、自然和流畅的感觉体验，而不是生硬的机器感。



Mac OS X的动态Dock给每位用户都带来愉悦的感受

不管是 Web 也好，Web-based 产品也好，我们都需要和正规、标准的操作系统一样，充分注重一致性的原则。如果你某个标题使用的是 16 号粗黑体，那么所有的标题文字都得是 16 号粗黑体。如果某个列表的显示内容是通过“上一页、下一页”来翻看，那么其他表格内容就不能通过“第 × 页”来选取。

我们可以先简要讨论一下界面设计的三个主要方面：菜单、图标和屏幕布局的设计。它们适用于各种交互式产品，而不仅仅是 Web 产品。其适用范围包括从标准的应用软件到掌上 PDA，甚至连微波炉也是。

3.4.3 菜单（或者导航栏）的设计

很多软件或网页都会用到菜单或者导航栏。菜单提供了与各种命令相关的选项，用户可以通过选择这些选项来执行与他们任务相关的操作。而导航则用于指示浏览者在 Web 产品中的位置，需要的话还可以分为二级导航栏甚至更多。

因此，它们有一个重要的共同点，就是需要根据任务的结构进行设计。与此同时还要注意所表达的信息。

菜单或导航栏有多种形式，例如下拉式、标签式、弹出式甚至对话式。在这里我不是要讨论如何把它们设计等更美观，而是如何让它们更具可用性，并且让用户满意。例如，永远都不要把一些对立的选项放置在一起。

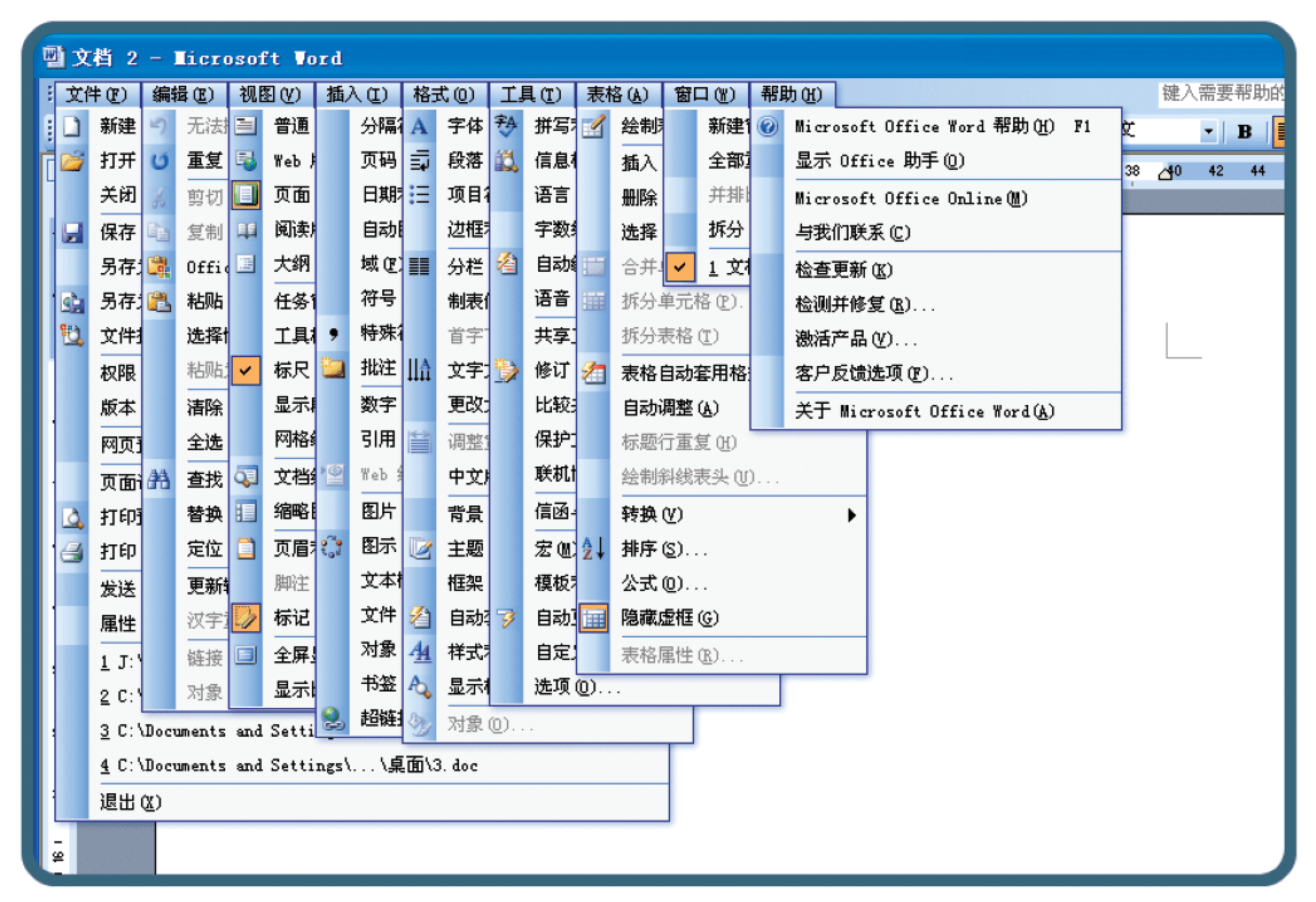


把这两个玩意分开

在设计菜单时，适当地进行分组是非常有必要的。ISO9241（交互设计的主要国际标准）对如何组织菜单选项进行了非常详细、客观的描述。比如说它规定，如果菜单含有大量选项（8 个或以上），就必须按照功能或其他对用户有意义的方式对选项进行逻辑分组。

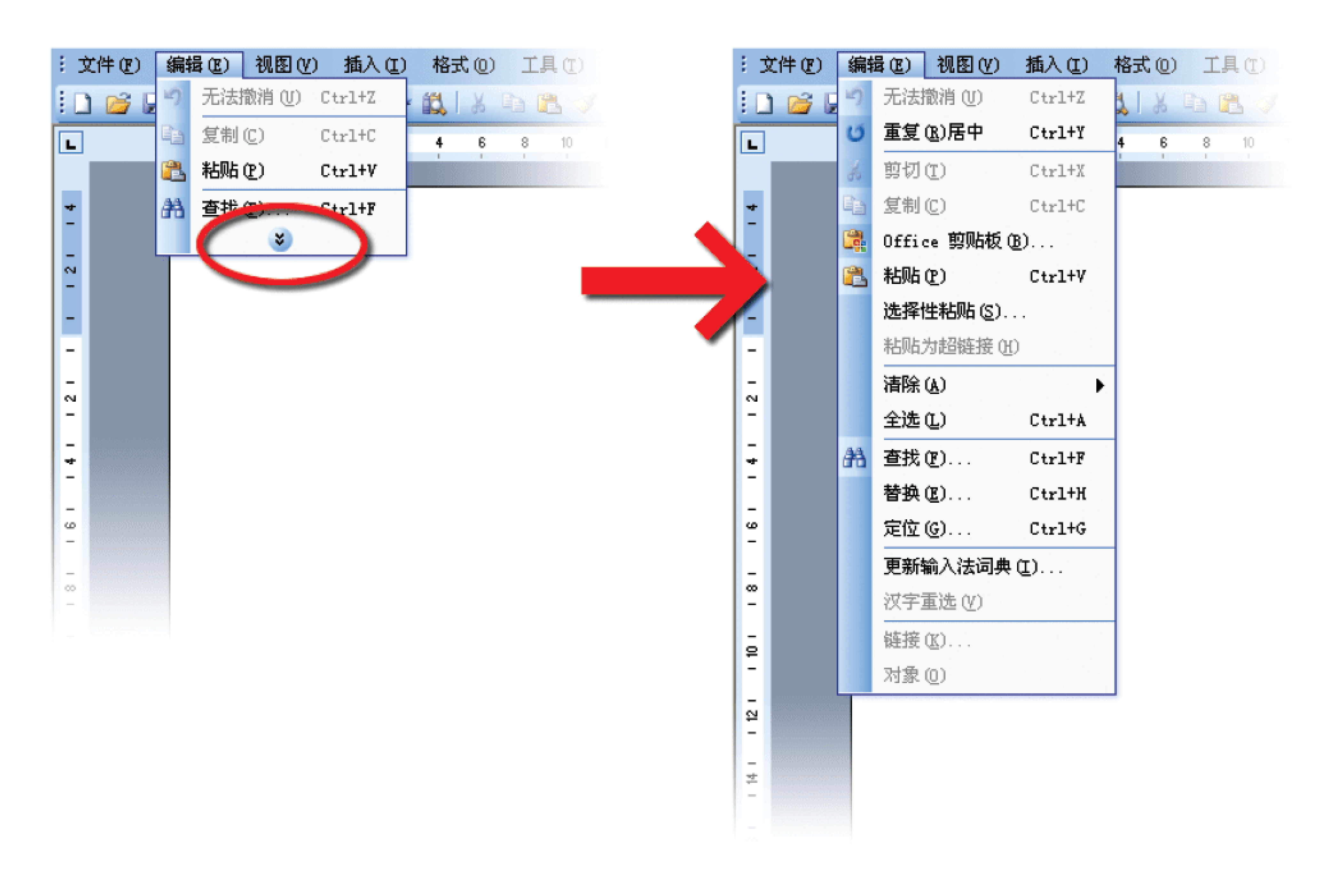
Office Word 的功能非常强大，菜单包含的选项超过了上百种。把所有的功能选项都堆在一起是无法想象的，用户需要好几分钟才能找到他需要的选项，这可能会让人崩溃。

所以现在你看到的 Office Word 将菜单分为 9 个组别。其中主要的标准是按功能分类，另外也会根据特殊的操作对象分类。例如，“编辑”和“格式”菜单组属于功能类别，包含“撤销”、“查找”和“替换”等操作；而“表格”菜单组则属于特定的操作对象，里面包含的选项都是针对表格的。

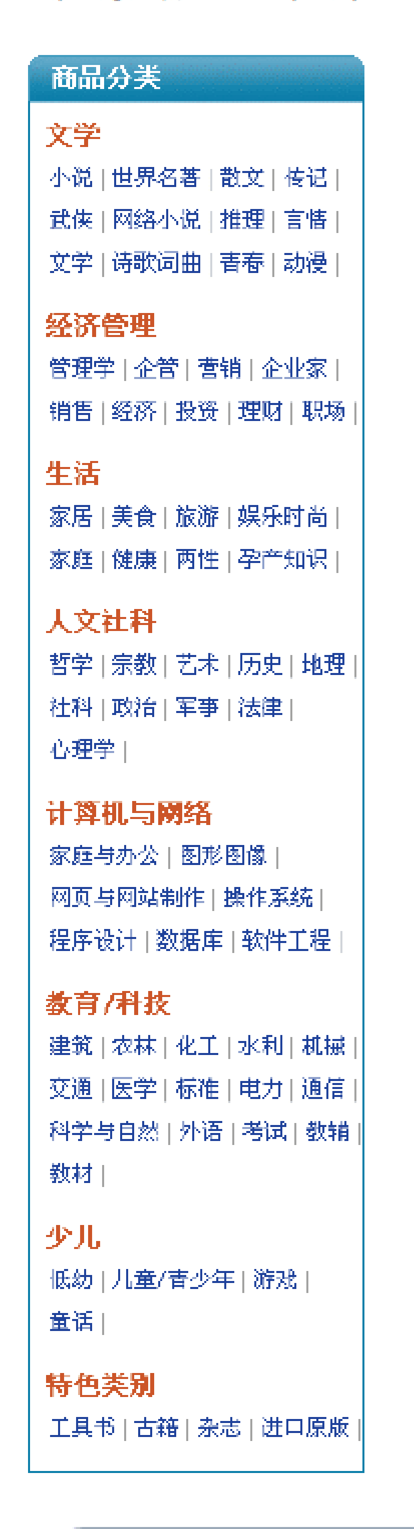


想想看，如果不对它们进行分组……

如果分组后选项还是很多，有时候你可能还得进行第二次分组。例如 Office Word 的菜单。这也是为了提高用户的识别效率。它还对不常用的选项进行了隐藏。而当用户需要选择那些不常用的选项时，也可以很快地找到。



在网站的导航设计中，你同样需要注意选项的组织问题。栏目的数量太多会明显增加阅读难度，所以大多数网站都会使用分级导航，也就是对栏目进行分组。

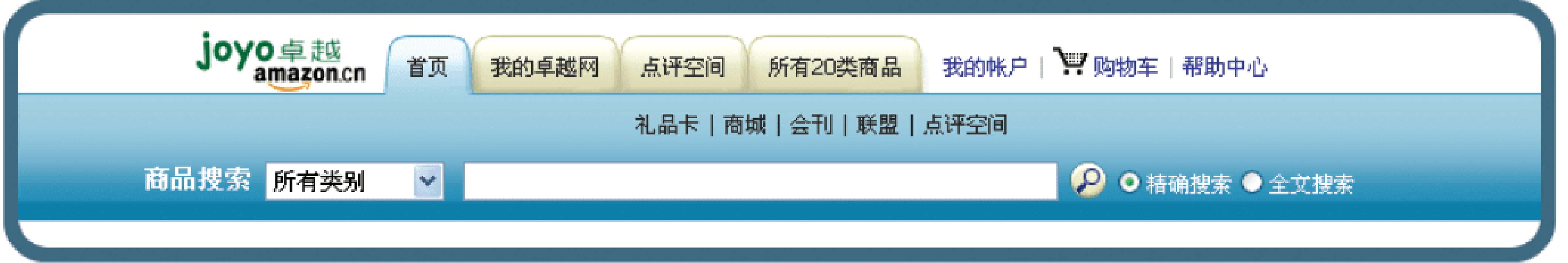


“卓越亚马逊”的导航设计则是另一种特殊的模式。这是一个信息量非常庞大的网站，包括图书、家电、电脑配件、玩具、厨具、家居、礼品等 20 个类别，每个类别又包含数十个子分类（看看旁边的列表，这还仅仅只是图书类别的分类）。很明显我们无法按照商品的类别来对导航进行划分。

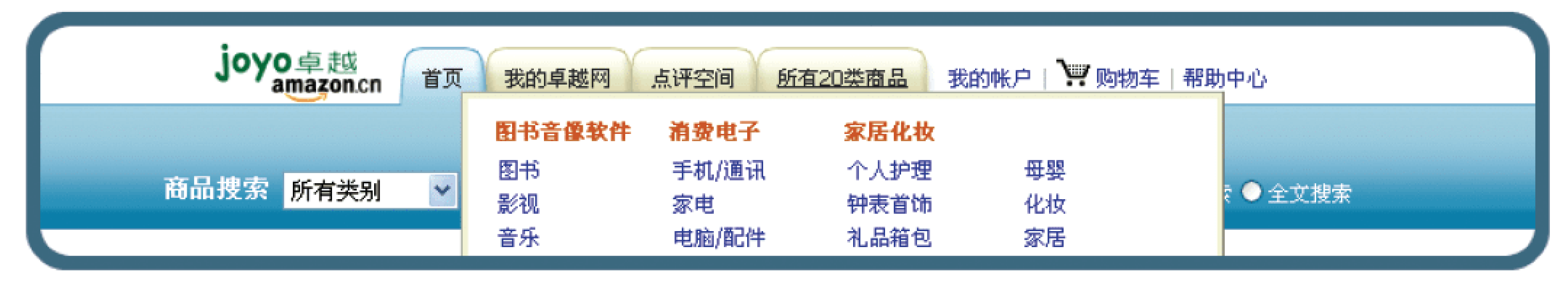
设计师们的思路是，当用户需要选择某类商品时，短时间内不会再去选择其他类别。因此大的类别只需显示一个，同时保证有选择其他类别的入口就行。

这种方案被称之为“动态 Tab 式导航”，也就是说，在缺省状态下并不直接提供产品分类，只有一个“所有分类”的标签。当你在“所有分类”中选择了某项产品后导航栏会增加一个标签（它取决于你所选择的产品类别），而用户切换产品类别时也只有这个动态标签发生变化。

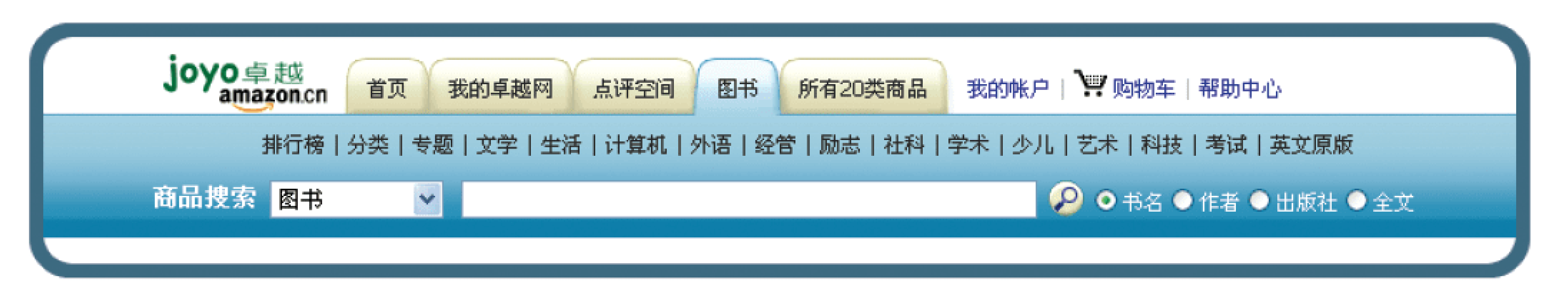
所以现在你光顾“卓越亚马逊”，在导航栏的位置会看到 4 个标签，并没有任何明确的商品栏目。这也促使你去点击“所有 20 类商品”的标签，从而能够很方便地浏览网站所有经营的项目并进行选择。



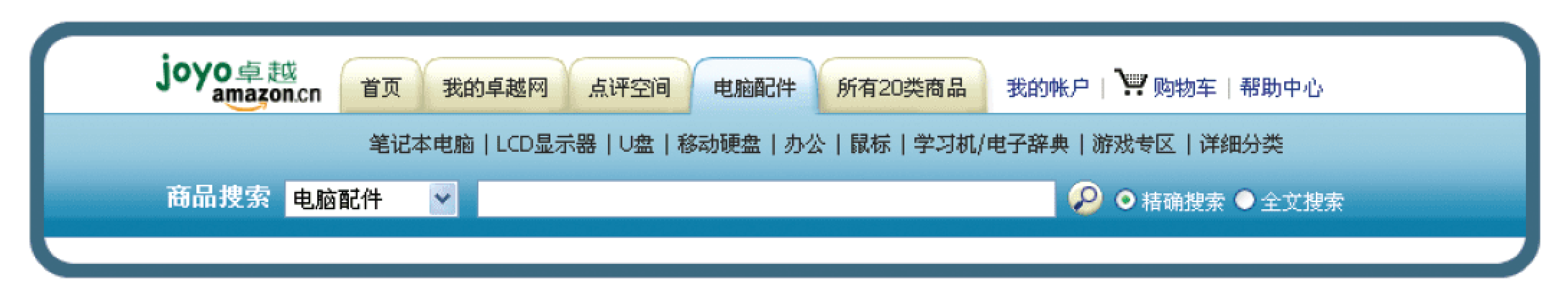
默认的导航菜单



点击“所有20类商品”的标签选择商品



选择“图书”后出现“图书”栏目和下属二级导航



选择“电脑/配件”后，“图书”栏目被替换为“电脑配件”和下属二级导航

此外，菜单项或者导航栏目应采用什么名称也很重要。由于显示空间有限，你不可能在菜单里面放上一句话来表明选项的含义。所以这种名称应当简短、明确，没有歧义，以免相互混淆而导致用户选择错误。例如在卓越亚马逊中，“最新的精彩评论和我的评分”这项功能就简化为“点评空间”作为菜单项，当你点击该标签时，将出现最近的评论列表，你也能提交自己的评论信息。

3.4.4　图标的设计——高度浓缩的图形

图标就是具有指代意义的图形符号，这是一种高度浓缩的图形，能够很快地传达信息，而且便于记忆。图标的应用范围很广，包括软硬件、网页以及各种公共场合，例如洗手间的标志、交通标志甚至是洗衣方法等等。它同时还与文化与使用环境相关。要设计一个好的图标可并不容易。



对于计算机操作系统来说，图标有一套标准的大小和属性格式。例如 16×16 像素和 32×32 像素，或者 256 色图标和 32 位真彩图标。它们都含有一定的透明区域，在透明区域内可以透出图标下的桌面背景。

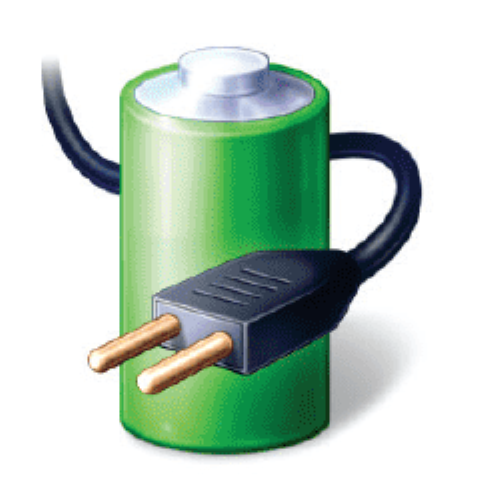
在很久以前（其实也就是几年前），用像素逐点逐点地进行绘制图标是最好的选择。这是因为当时的计算机性能并不高，使用像素可以节省图标的体积，也便于快速地读取。早期的操作系统使用的都是像素图标（包括现在大部分手机、MP4 等便携设备）。例如这样的：



别以为像素图标都像Windows里面那样死板板

当然这样的也很有魅力。但是现在的操作系统平台所能支持的图标格式（例如 PNG）已经可以实现多通道透明渐变了，也就是说，你可以给图标加上阴影甚至模糊的效果。而在 Windows Vista 出现后，PC 支持的图标尺寸已经达到了 256×256 像素。包括我在内，现在很多人都使用 Adobe Photoshop 或者 Illustrator 来绘制大尺寸的精美图标，而且能够直接通过缩放得到不同的尺寸（像素图标就不行了，不同的尺寸都需要设计师一个点一个点地逐一绘制）。







Windows Vista操作系统里面的部分图标







Mac OS X操作系统里面的部分图标








我模仿上面风格绘制的一些图标

对于 Web 或 Web-based 产品来说，图标只是图片而已，它的格式可以是 GIF、JPEG，也可以是 PNG，总之只要能够在 Web 浏览器中显示就行。所以你可以随意地使用像素或写实的风格，达到自己所想要的任何效果。但美术设计不是这本书所要讨论的内容。要想设计出优秀的图标，你至少需要注意以下几点。

不要“辞不达意”

图标的图形必须要适合所代表的意思。最理想的情况是，在没有辅助说明文字的条件下，用户仍然能够通过图标来理解它所代表的功能或概念。



Google产品的图标排列

很明显，上图中绝大部分的图标都能够贴切地反映相关功能，例如“大学搜索”、“地图”、“移动服务”、“Google 实验室”和“网页搜索”。但也有少数几个的含义不那么明显：“工具栏”、“翻译”和“网页目录”。也许你通过旁边的文字能够明白代表“工具栏”和“翻译”的图标含义（例如“翻译”是把一团乱麻整理为清晰的内容），但是“网页目录”的图标就不是那么回事了。在这里它的功能更多偏向于美观方面。

设计代表实际对象的图标相对要容易一些，因为你可以直接使用它们的图像，例如用房屋表示“主页”。设计表示动作的图标可能会有些难度，但也不是没有可能。你可以通过逻辑上的联系来进行设计，例如“剪切”的动作可以用剪刀的图像来表现，“加密”使用锁或者保险箱、“查找”用放大镜或望远镜等等。



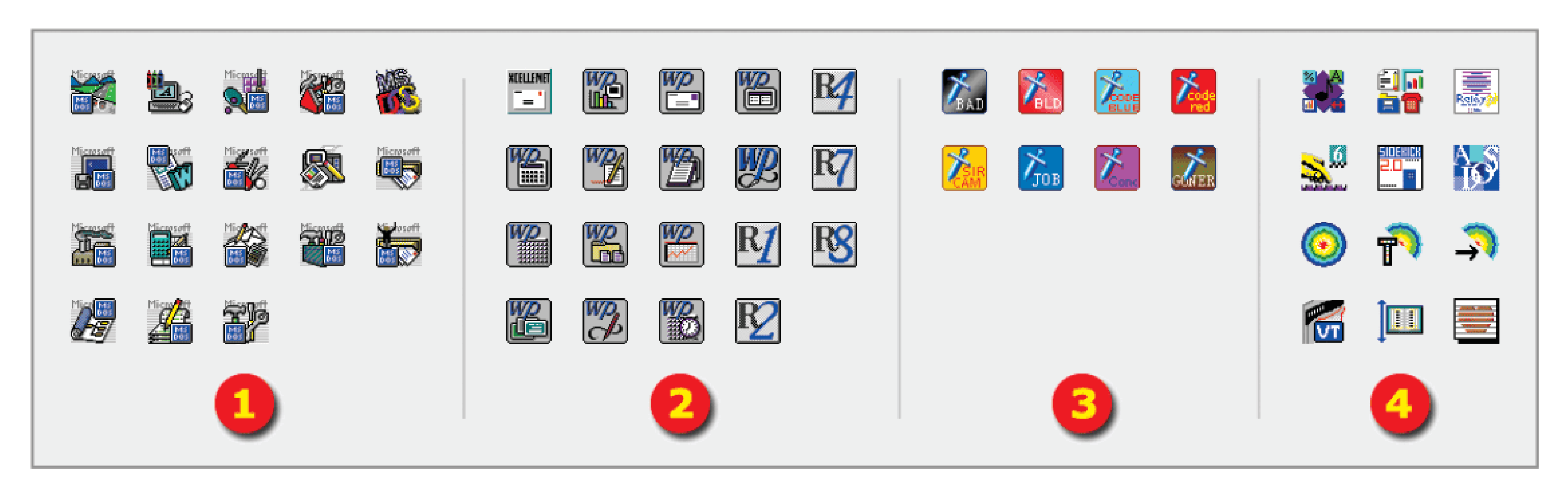
用静态图像表现动作的概念

小而简单

图标应当小而简单，应该能让用户一目了然。它们毕竟是辅助的元素，所以不能设计得过大，占据显要的位置。另一方面，Web 产品本身的特性也不太赞成我们使用大面积的图片，因为那样会延长网页显示的时间。

“一目了然”的意思是，你需要让用户能够理解它们的含义，并且能够毫不费力地区别不同的图标。这个道理很简单：如果每个图标看起来都差不多，就仅仅只是图案而已，没有任何的辨识性和指向性。在这种情况下用户只能根据文字来辨识功能。

看看下面这些图标，我真的很想知道它们都代表什么意思……你能分辨出来吗？



一些早期的图标……或者说被淘汰了的图标

3.4.5 屏幕布局的设计

屏幕布局的设计，就是把页面或界面所需要显示的内容进行分组和排列，使之更具可用性。

大的方针

在你设计界面或者网页页面的时候，应该避免两种极端的情况：

1. 屏幕过于稀疏，含有太多空白的空间

2. 屏幕过于密集，含有太多复杂的信息

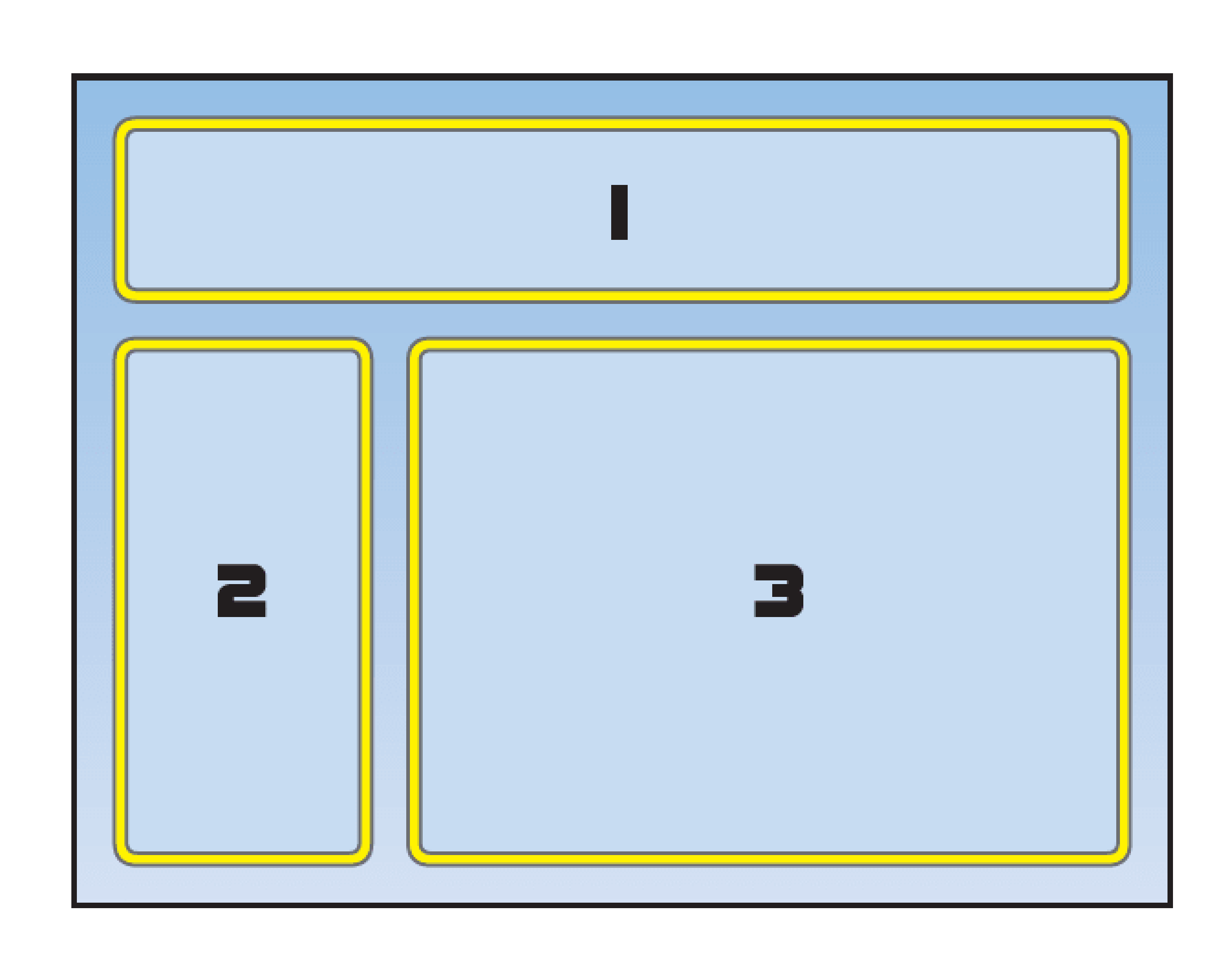
有时候设计师可能希望通过大面积地“留白”来增加页面的通透性，或者让页面显得更“高雅”（就像很多书籍或平面设计那样）。使用空白也有助于突出信息。但是如果空白空间太多的话，单

个页面中显示的信息量必然会减少，用户可能需要频繁地上下拖动网页，或者切换界面屏幕，因此他们的交互过程会被经常性地打断。

而把信息都集中在一起则是另一种极端，页面上到处都是密集的文字、色块、图标或图像，不停地分散用户的注意力。他们很可能完全不知道该从哪里下手，也容易造成混淆。

对于网页设计来说，你还需要考虑其他问题。在使用网页的时候，用户应该随时都能了解以下三个问题的答案 ：“目前我在哪里”、“这里有什么内容”和“我还能够去哪”。这能让他们建立良好的位置感。

最简单的解决方案，也是目前最常见的网页设计方式，就是把网页划分为三个区域。如同在下面这幅图中你所看到的 ：



在上面这幅图中，顶部的区域也就是**区域 1** 回答了“目前我在哪里”这个问题。在很多情况下，用户并不是通过主页进入某个网站的，尤其是在搜索引擎出现之后。他们就这么突然地、没有敲门就进来了，而且是蒙着眼睛进来的。而等到他们摘下眼罩，你就需要让他们知道身在何处。所以在这个部分你需要放置网站的 logo 标识和导航栏。

左侧的**区域 2** 则是页内的导航，也就是当前这一层的所有内容。这里多是一些链接或菜单，便于用户了解本网页还提供了哪些内容。也就是说，这个区域让用户能够看到在这个子栏目中还有些什么他们可能会感兴趣的东西，并且提供给他们明确的入口。这就回答了第三个问题 ：“我还能够去哪”。

而最重要的信息放在**区域 3**（或者说内容区）。它其实就是用户访问网页的目的。这个区域回答了“这里有什么内容”的问题。用户一般都只是大致地浏览，如果不感兴趣他们可能马上就离开。所以网页的内容应该简短、明确，使用标题或者分组来突出要点或者划分段落是个比较好的方法，

有助于用户捕捉重要的信息。

看看下面这个网站，很明显的就是按照我们刚才所说的三个区域进行划分的：



当你通过某种途径来到这个页面时（可能是 Google“微软雅黑”得到的一个超链接），区域 1 很明确地告诉了你现在身在何处：优艾网（同时你还知道了这是一个关于人机界面的“中国专业门户”）的“人机交互”版块。

而区域 2 则提供了这个版块内部的一些其他相关文章的入口，如果感兴趣的话，你可以很方便地前往这些页面进行阅读。

区域 3 是你来到这个页面的本来目的。它是这个页面的主要内容区域，其中最明显的是“微软雅黑 VS Windows 宋体”这篇文章。很可能它就是你想要的——了解有关“微软雅黑”的知识。

同时也要考虑细节

在页面布局的设计中，你也需要用到一些视觉交流的原则。例如重要的信息应该突出显示。同时，使用色彩、动作和分组的方法可以使界面结构更为清晰和易于理解。你的目标是让用户在第一眼看到界面的时候就能把注意力集中在对他有用的信息上。

有些设计师喜欢在页面中放一些动画或动态效果。这很好，但不能太多，否则会分散用户的注意力。你可以在高级原型中询问用户得出具体的结论。

有时候“分组”是很有必要的。这很简单，但非常有效果。看看下面这两个表格，第一个没有进行分组，而第二个则根据内容的相关性进行了一定的划分：

故障查询								
○快速查询	◉高级查询							
首次发生时间	从	MM/DD/YYYY	HH:MM:SS	到	MM/DD/YYYY	HH:MM:SS		
最近发生时间	从	MM/DD/YYYY	HH:MM:SS	到	MM/DD/YYYY	HH:MM:SS		
故障关闭时间	从	MM/DD/YYYY	HH:MM:SS	到	MM/DD/YYYY	HH:MM:SS		
故障源IP地址				□模糊查询				
故障描述				□模糊查询				
设备编号					故障级别	=	全部	
设备名称		□模糊查询			故障类型	=	全部	
故障采集源	全部				故障发生次数	=		
故障状态	◉全部	○当前	○历史					
结果显示格式	-							

没有分组的表格



进行了分组的表格

你可以看到，尽管所有的内容都一样，排列顺序也没变，我只是去掉了几根分隔线而已。但效果很不错。

还有很多地方需要注意

在关于屏幕布局的设计中，还有一些关于可用性所需要注意的地方。例如，在设计中要把握好页面所包含图片的总“体积”。在网页中使用图像能增加吸引力，但如果图片过多就会影响到网页下载的时间。

另外元素的“一致性”也是个问题，例如网页中的“超链接”。传统的表示方法是蓝色文字加下划线，但现在的网页设计更为灵活，也更加个性化，很多网站都不再使用蓝色加下划线的表示方法。有些超链接不带下划线，有些带下划线的文字也未必能点击。

这种做法没有问题。但是要坚持一点，就是你必须坚持“一致性”。举个例子来说，如果你的文章列表需要入口，那么你可以使用“>> 更多”，也可以使用“>>more”。但不能有的地方是“>> 更多”，有的地方又是“>>more”。

看看下面这个网站：



注意左下方的“>>更多”是黄绿色

它的问题是，几乎所有的“>> 更多”都是深灰色，但左下方“专题”栏目里却是黄绿色。这时一个问号出现了：“为什么这里跟别的地方不一样？”我看了看所有的文章列表，更多的问号出现了：“专题非常特殊吗？”—“如果是，为什么这个版块并不是特别突出？”—“如果要引人注意，似乎黄绿色也不是个显眼的颜色……难道我的辨色能力有问题？”……

我发现我的思路已经从看文章变成了怀疑自己的生理能力。绝大多数用户应该都不太喜欢这种体验。

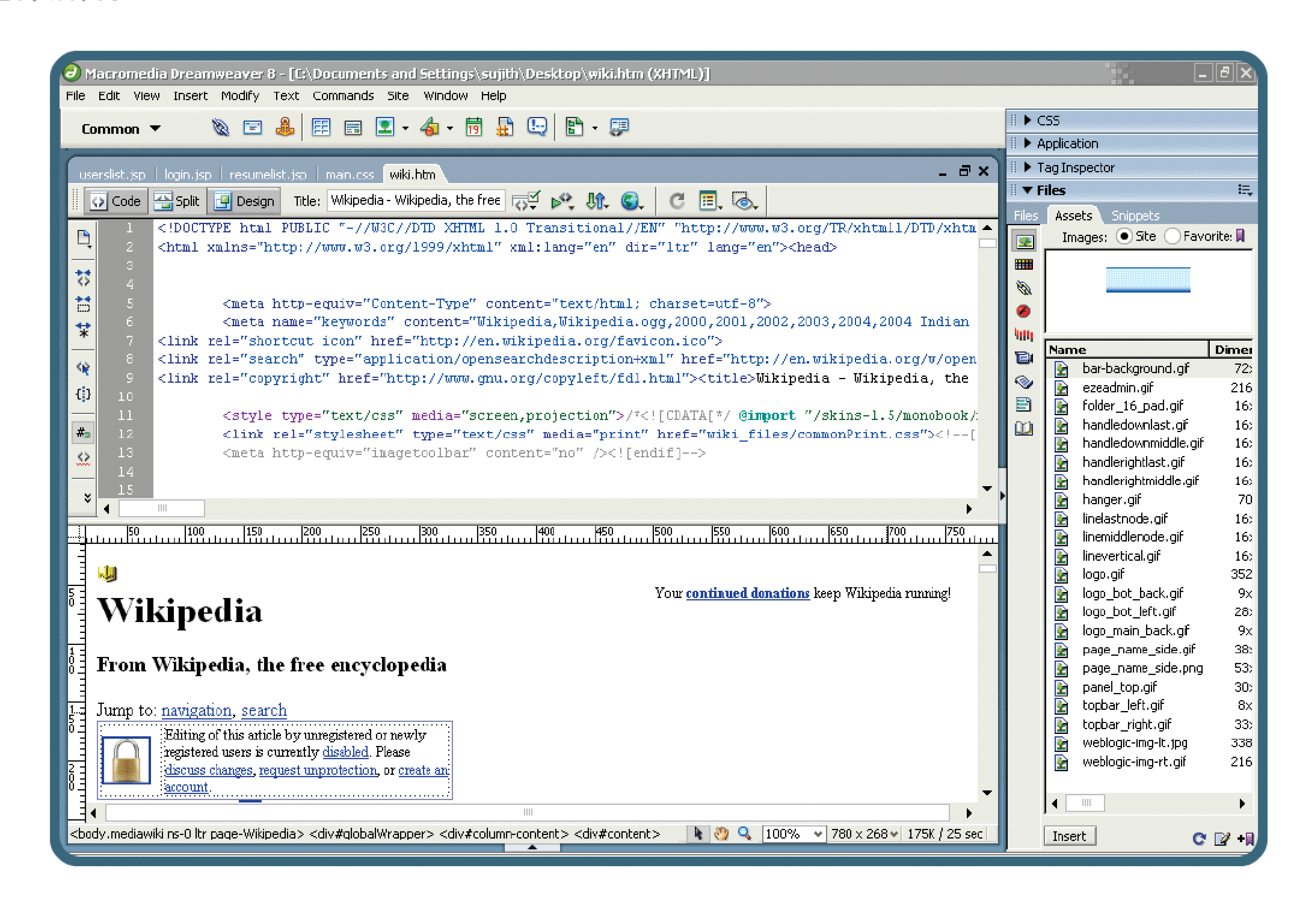
关于 Web 和 Web-based 产品的页面设计，在后面的章节我们还会进行更为详细的讨论。

3.4.6　别理那些复杂的工具，我们做的是W eb

不管是概念设计还是物理设计，你都需要一些界面设计的工具来制作原型或者开发正式的产品。

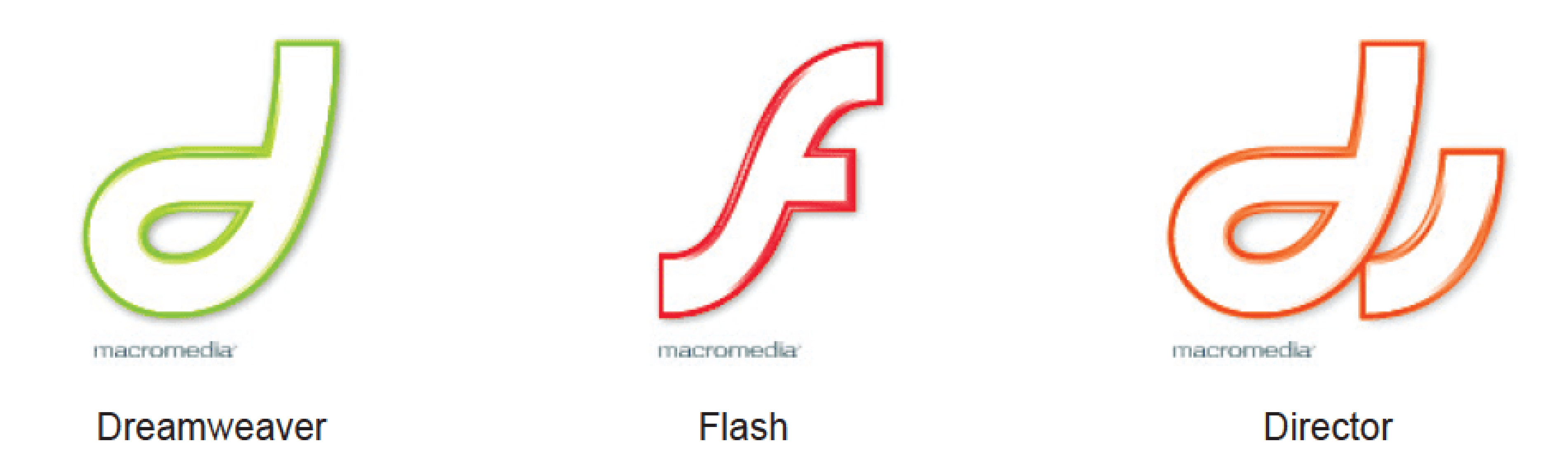
在这里讨论特定的工具没有什么意义。而且你总能看到新的工具出现，那些老的软件也经常推出新版本。可能在拿到这本书的时候你又会看到或使用到很多新的设计软件。

对于 Web 和 Web-based 产品而言就更简单了。你设计成品时用的是 Dreamweaver，制作原型同样也可以使用它。这种软件大多数都是面向对象进行设计的。当然你也可以直接对程序代码进行编辑。



你应当了解 CSS，它能灵活地进行设计，并且严格地控制页面格式，还能序列化所有的内容。另外 JavaScript 也很有用。

还有一些面向对象的软件是 Macromedia 公司的 Director 和 Flash，它们同样能够很方便地开发出包括高级原型在内的各种交互式产品。当然我更为推荐的是 Flash，我一直使用它。



Dreamweaver　　Flash　　Director

第四章

以用户为中心的开发



“谁跟你客气了！我是正面人物！主角！”

“但是这个主角啊……不是谁都能演的。”

——小品《主角与配角》

终于到正式开发的步骤了。

别以为之前都是在做无用功，或者认为那只不过是“做做样子”或者“走走过场”。只有完全弄明白用户是谁，他们想要什么，你才能开发出最适合他们的产品。很多人一直只是在开发自己认为好的产品。但是他并不能代表所有人，尤其是那些没有技术背景的用户。这种开发的结果可能就只有他自己满意，却没有其他人认同。

你不能用“曲高和寡”来作为托辞。因为产品开发出来后的用户并不是你,或者不仅仅只有你。用户决定着产品最终的评价，如果他们觉得你做得很臭，那么就说明确实很臭。世界上所有的商品都是这个道理——顾客不能接受的东西就卖不出去，那么生产商就要吃亏。

所以产品应该想方设法地“取悦”用户。而要想顺利地取悦他们，就得深入地了解他们。这就好像如果你想获得心上人的芳心，就得事先做一些了解，看看她喜欢什么，讨厌什么。如果她对海鲜过敏，那么就不能请她去吃龙虾。

我一直都在强调“以用户为中心”的重要性，我如此多地使用这个词，以至于我的输入法都记住了。

4.1　为什么要让用户掺和进来?

“了解用户及需求”和“制作原型供用户评估”都是“以用户为中心”的具体体现。我也介绍了一些数据搜集的技术，这些技术能帮助你透彻地了解用户和任务，从而指导后面的设计。很明显它们都和用户分不开。

而“以用户为中心”开发的另一个特点是**让用户直接参与开发**。

让用户参与开发有两层意思。第一层意思是你来开发产品，让他们参与产品的测试和评估；第二层意思就是请他们积极参与设计本身——也就是成为合作的设计师。

关于让用户参与产品测试是非常有必要的，因为这是一个检验的过程。在第八章我会专门讨论这个问题。而让用户参与设计，你就能随时了解他们的想法，让他们跟着开发进展一步步地走，出更多的力。这样最终的产品自然也会更加成功。

另一方面，让用户参与开发还有两个原因。这两个原因与产品的性能没有多大关系，但同样非常重要。

告诉他们别太天真

有时候用户对新产品的看法和期望不切实际。在销售人员把产品吹得天花乱坠之后，他们可能

会把它想象得无比美好，认为从此天更蓝，水更清。但是，期望越高可能最后的失望就越大。当他们发现最终产品出来后也就那么回事的时候，就会感到受了“欺骗”，因此很可能会抵制甚至拒绝接受你的产品。

所以你应该确保最终的产品不会出乎他们的意料之外。如果让用户参与开发的整个过程，他们就能在第一时间了解产品的能力大概是个什么样子，也能更好地理解对自己的工作会产生多大的影响。他们也能理解你为什么要这样或那样设计产品。如果在开发过程中产品与最初的承诺出现了偏差，他们也能理解这些改变都是有原因的，相应的也会随时调整自己的目标。这些都有助于避免最终产品让人失望的情况发生。

另一方面……他们才是真正的主人

另一个原因是让用户对产品产生“拥有权”。让他们参与设计工作，见证开发的辛苦，并且感觉到自己对产品的完成有一定的贡献，这会使他们认为这个产品是自己的。这样就能更容易地接受它。

人们总是渴望自己的意见得到重视。如果你让用户真正地参与进来，他们会觉得自己对于这个产品非常重要，因此也更愿意花时间与你合作。

在交付产品之前，让用户开发和使用产品也能够增进他们对最终产品的理解和认同度。很多东西并不是仅仅通过培训或演示就能达到的。付出了心血和劳作（有时候可能是繁重的劳作）之后，他们就会对自己参与开发的产品产生感情，这是我们最希望看到的。

4.2　支持用户，而不是限制他们

所谓“UCD——User Centered Design”，指的是你应该以用户和他们的目标作为产品开发的驱动力。良好的设计应该支持用户，而不是限制他们。

这是一种设计思想，而不是纯粹的技术。

要支持用户，以下有几条具体的原则：

排第一位的永远是用户，不是技术

记住你不是技术的奴隶，用户也不是。你要问的问题是：“为了更好地支持用户，我应该选用什么技术？”而不是“这项新技术很酷，我想应用到这里。”再新的技术也只能是为服务所用，不能本末倒置。





给他们最习惯的环境

这也是旧话重提。不光要了解用户的任务目标，你也一样需要了解他们是如何执行任务的。这种对应用环境的研究能让你找到一些不易发现的问题。用户在他们最习惯的环境中工作是最理想的，因为不需要再去学习。但这不是说要把那些不好的习惯也照搬到产品里面去。

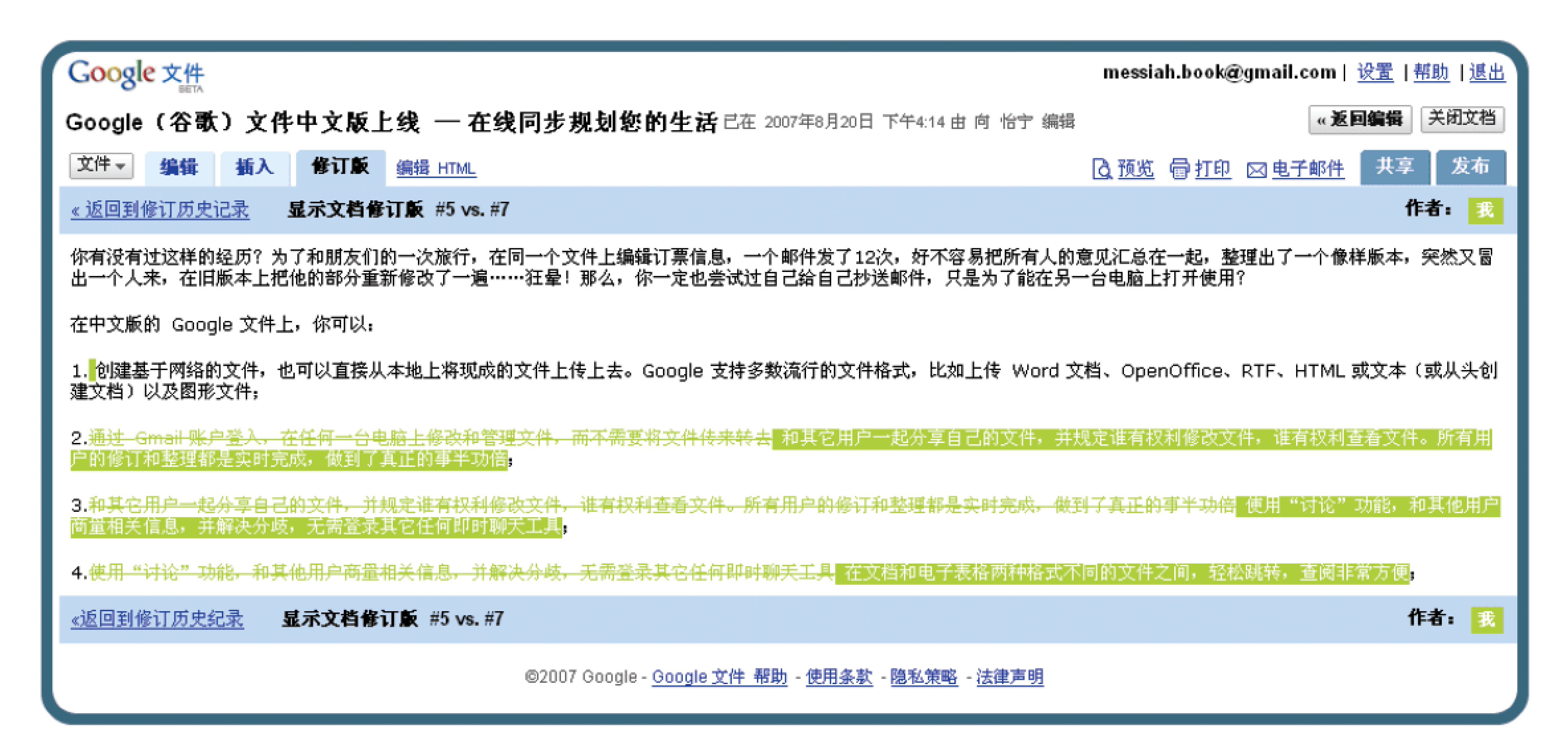
要支持用户，就得考虑周全

人们总是很容易犯错误，而且不同的人也会有这样那样的限制。也许有的人记性不好，有的人老喜欢走神，还有些人连英语的“Yes”或“No”都分不清。如果想支持用户，你就必须考虑这些限制，从而降低用户犯错的可能性。

经常向他们咨询意见

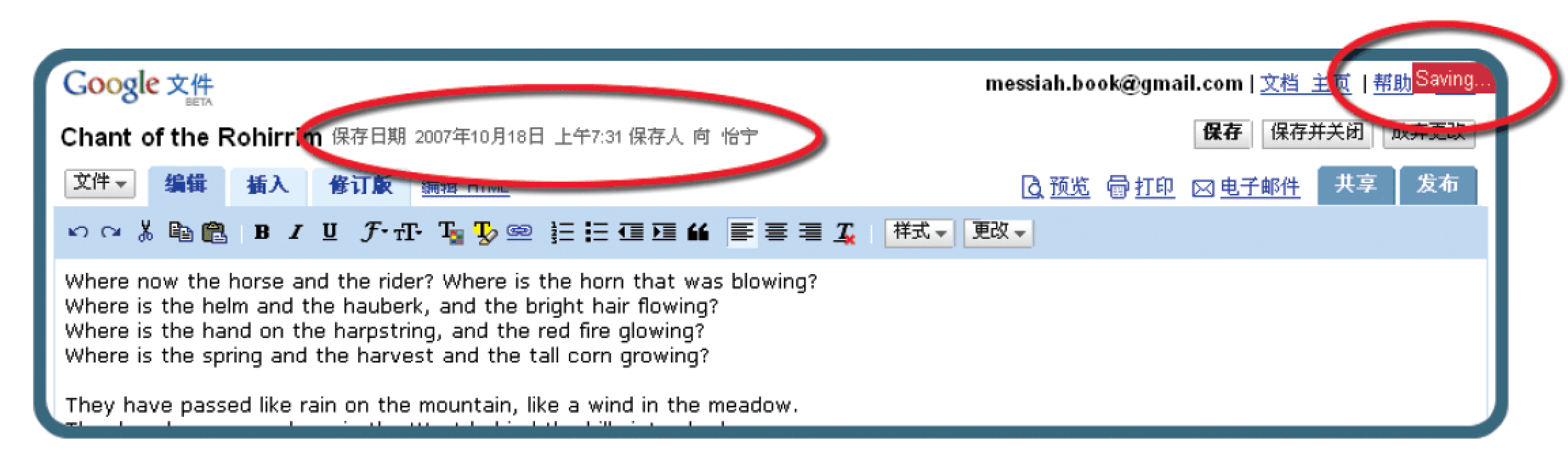
不管用户以何种形式、何种程度参与开发，你都应尊重他们的意见，这非常重要——哪怕他们有时候提出的意见非常幼稚甚至毫无逻辑。你可能永远无法清楚地了解用户的想法——但没有他们的认可，你的工作就没有价值。

“Google 文件”就极大地支持用户而不是限制他们。这种“只要是相关用户，无论他们在何时何处，只需找到最近的计算机登录就可以编辑共享文档”的服务方式，无疑是通过研究用户行为和应用环境所得到的结论。与此同时，你还可以对所有用户更改的历史记录进行比较，这明显改进了用户的工作。



针对同一文档的各历史版本进行对比

Google 的设计师们同样注意了避免用户无心犯错的一些地方。例如人性化的定时保存功能，你再也不用担心突然遇到停电或死机，可以高枕无忧地编辑文档。系统在默认情况下三分钟保存一次，并会显示保存日期和时间。



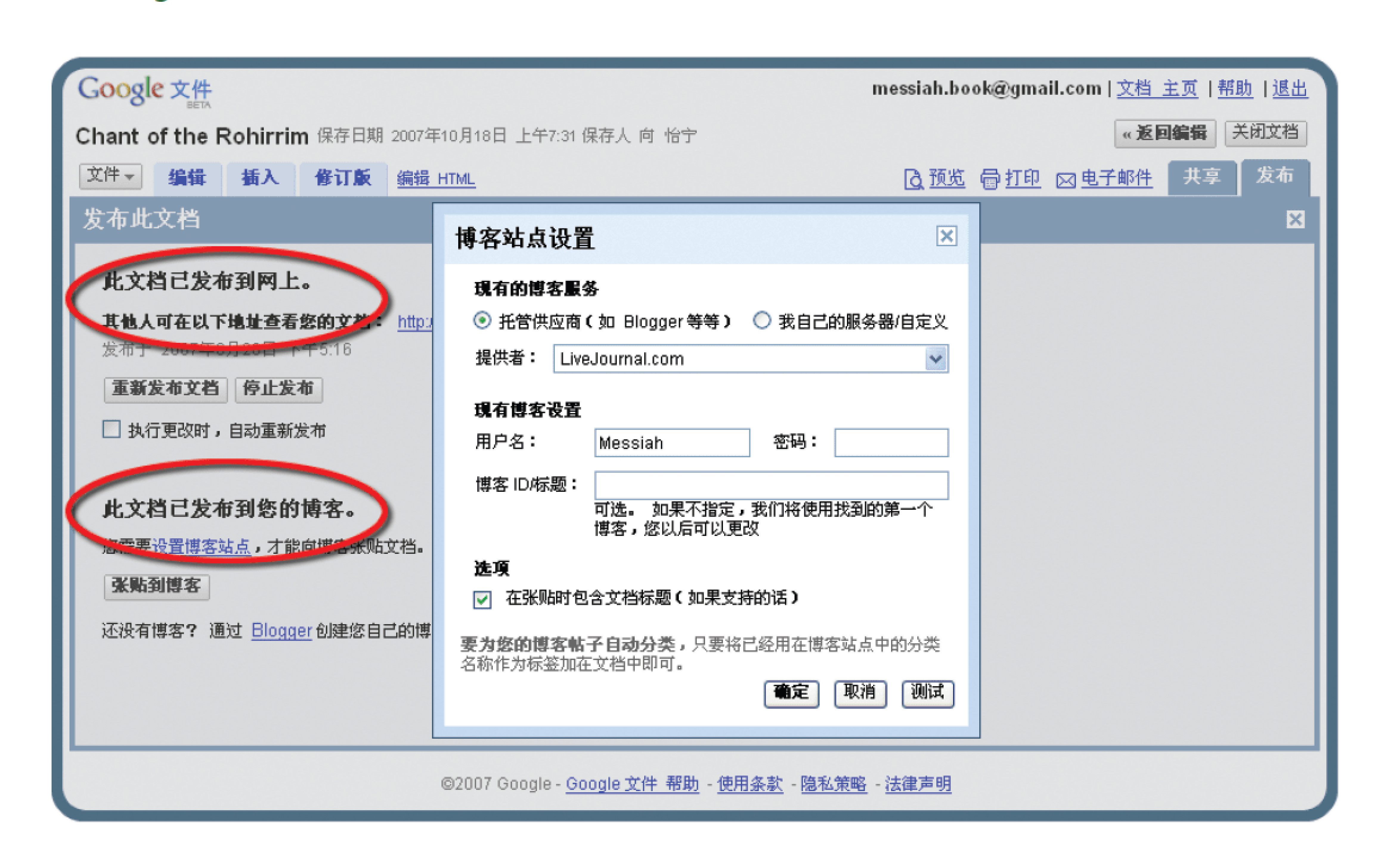
我没有点击“保存”，但Google帮我做了

同时“Google 文件”还具备多人协作、公开发布和网络共享功能。所有者对文件具有所有权，也能邀请更多的协作者和浏览者加入。协作者同样可以编辑文件，而浏览者可以查看文件的最新版本，但无法做出更改。



共享文件。作为文件的所有者，你可以有更大的权限

“Google 文件”还支持将文档发布到网上，甚至是你的博客。



我相信上述这些功能都是开发小组在充分理解了用户特征、明确了用户需求后所得出的。例如那些拥有自己博客的“特殊”目标用户。虽然“Google 文件”提供的功能选项没有 Office 丰富，

但足以应付我们日常的办公需要。这些功能极大地支持了网络用户。

如果你有兴趣，不妨尝试一下“Google 文件”这样一款在线办公软件（网址是 http://docs.google.com/）。它确立了一体化平台，集多种服务于一身。我不是在做广告，Google 也不会因此给我什么小礼物。我仅仅只是想在可能的情况下你完全可以尝试一下与Office不同的文档处理系统，看看它们之间的差异和各自的优缺点——不过有一点：“Google 文件”是完全正版免费的，你在使用它的时候腰杆子完全可以更直一些。

4.3 如何让用户参与?

可能很多设计师并不太希望那些“用户”突然出现在设计团队里面：生疏的面孔、与设计师迥然不同的工作方式、或小心翼翼（当他们觉得设计师很神秘时）或专横武断（当他们觉得自己很重要时）的态度、以及那些层出不穷的问题（要么就是他们一窍不通，要么就是他们过于投入）……

这有点类似奇幻小说里面的精灵和矮人种族，他们天生就是死对头——“我们可以在一个地球上，但是天天在一起工作……我不太确定会有什么后果！”

让用户参与开发确实可能令人不太适应，但是它的重要性和好处我想你也知道。而且我认为你可以通过一些手段来避免两个“种族”合作所产生的种种麻烦。另外，用户参与设计也有不同的形式，你可以针对不同的状况来决定让他们参与的程度。

4.3.1 用户参与的不同形式

关于让用户参与开发的程度，我们不如先看看两个极端的情况，然后你可以根据实际情况来自由掌握。

极端之一：让用户作为设计组成员

也就是把用户当作“自己人”。在有些情况下，用户可以全职或兼职加入到设计团队中来。如果他们全程参与项目，这可能是一种理想情况：熟门熟路，配合默契，同时他们也能够透彻地理解系统及其原理。

但是，如果项目周期太长，例如需要几年的时间，这些用户就有可能逐步“蜕变”成完全的开发者。他们会与其他那些“真正的用户”失去联系：不再熟悉产品的使用环境，也不再了解现在的需求。这样一来他们的见解就不再那么有价值了。



如果用户是兼职参与项目的开发，可能情况会好一点。但是你又会面临一个新的问题，就是如何管理和控制他们。这些可怜的家伙身兼两职，他们既需要学习新的术语和知识、处理不熟悉的材料，还得完成自己本来的日常工作。这种压力可能相当大。

也许可以让不同的用户轮流兼职参与开发，例如每 3 个月更换一批参与设计的用户。但是这又面临着不同用户之间的差异性。

极端之二：用户远距离参与设计

另一个极端，是让用户仍然做自己的工作，只不过定期通过邮件或报告的方式接收有关项目进展的信息。偶尔会集合他们对产品进行讨论。这也是一个有效的方法，同样能够使用户对产品有“拥有权”，同时也不会出现“最终产品无法满足期望”的情况。

但是这种方法的效率有些不够高，那些时间紧迫的项目就更难说了。

有些头疼，嗯哼？

用户与开发工作距离的远近，需要你针对具体的项目进行具体分析。比如说，如果存在大量的用户（如成千上万的用户），你就可以采取折衷的方法，让各个用户组派代表以全职的形式加入设计团队，而其他的用户则参加集体讨论或者其他数据搜集的活动。

另外，如果用户比较容易确定，例如为某一个公司开发的产品，那么让用户参与就比较容易，因为很明显该公司的员工就是用户。而且因为他们需要这个产品，相互间的协作也更为方便，完全可以配合开发工作借调员工来参与设计。但如果开发的是面向开放市场的产品（例如游戏网站），就不太可能让用户作为设计小组的成员。

4.3.2　用户参与的是与非

“以用户为中心”的方式是让用户参与开发，从他们的角度来考虑设计，最终达到他们的目标。但是，这并不代表着你得“一切都听用户的”。这是一个辩证的观点。

最正确的态度应该是：**“别只相信技术，也别只相信用户”**。用户并不是专业的开发人员，所以有时候他们提出的意见只能作为参考。你不能完全按照他们的意思来。这是因为用户可能会在参与设计的过程中表现出以下特点：

- **他们可能会畏缩不前**

  有些用户并不是以“主人翁”的姿态参与开发，这是最让人失望的地方。并不是说他们很傲慢，相反，他们过于拘束。而且很多人受限于一个传统的观点：“这是技术人员的工作，而我只是个用户而已。”有时候让他们做出决策是一件很困难的事情。有一次我给一位用户两种执行任务的方案，希望听听他觉得哪种比较好，他沉默了一会儿然后说：“我没什么意见，你给我什么方案我就用什么方案。这完全由你决定。”

- **他们没有系统开发的背景**

  这是一把“双刃剑”。有时候这是好事，因为他们不会被技术的框框给限制住，所以可能愿意接受新思路。但不了解开发常识又可能导致他们思路狭窄，习惯于自己熟悉的使用方式，哪怕很难用。如果他们已经尝试过或拥有过一个系统，那么新系统往往很难有所突破。

- **他们的意图不可预测，也分不清主次**

  有可能他们在初期没有什么主张，但随着项目的进行开始“灵感乍现”，变得过于“投入”，夸夸其谈。他们很可能会在即将要结束设计的时候提出很多复杂的想法，要求对产品进行重大的改动。这就意味着要把现有成果整个推翻，所有努力付诸东流。这很令人哭笑不得。

- **他们很容易担惊受怕**

  有些用户可能很喜欢这种参与性的工作，不用动太多脑筋，却又能被人当作权威来征求意见。这种心态可能导致他们“害怕”受到不好的评价甚至丢掉这份工作，所以不管是好的意见还是坏的意见都不敢提出来。有时候由于担心增加自己的工作负担，他们也会变得比较“中庸”。但这不是我们希望看到的。



两个用户的对话

- **用户期望值可能过高**

还有一些用户是“唯美主义者”。而且他们也不了解开发工作的困难程度，所以很多要求过于“严苛”而不够“宽容”。在技术可能无法实现，或者即使实现也需要非常高的代价时，他们总是无法理解你“为什么做不到”。这会给我们增加很大的压力。

以上这些都是很有可能发生的情况。用户不主动不好，用户过于“主动”也不好。但不管怎么说，你仍然有必要让他们参与到开发过程中来。用户需要正确的引导，你也需要甄别他们的意见哪些是积极而重要的，而哪些又是没有很大帮助的。这需要在实践中逐步领会。所以，你必须正确地面对用户，同时保持清醒的认识。

4.4 别用望远镜，要现场研究

只是把用户请到办公室里面参与设计还不够，你还需要“现场研究”。也就是说，你需要深入到用户的工作环境中去，观察他们如何执行日常的活动。作为观察者，你应该专注于用户所处的环境，尽量多参与他们的日常工作，不管是交谈、会议还是阅读文档。

坐在自己的办公室里，只用望远镜远距离地观察他们是不够的。

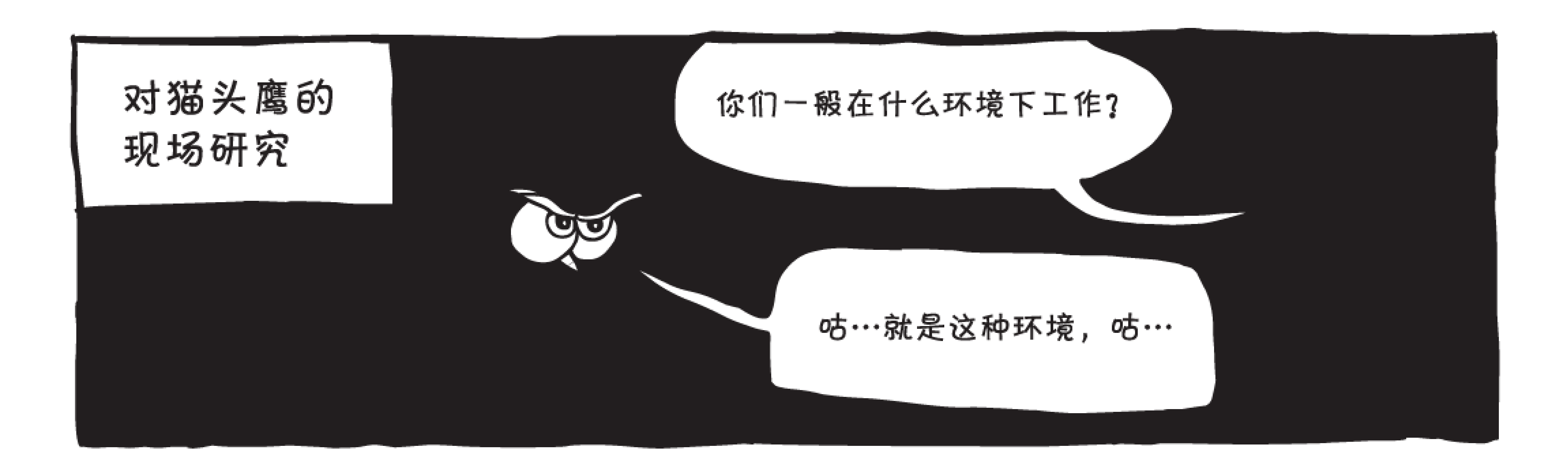
4.4.1 关于现场研究

用户们习惯了自己日常的事务，认为所有的东西都是理所当然的，所以往往会忽略一些事务

的重要性。如果不是在自己熟悉的环境下，他们可能根本就想不起来这些事情。

现场研究的目的就是要揭示这些隐含的东西。在与用户合作的过程中，不要和他们保持距离，你应该深入到他们的日常工作中去，研究他们工作的环境，观察他们如何执行任务，在过程中又遇到了哪些不便。近距离地观察用户能够让你了解他们真实的工作过程（有时候并不像他们故意表现的那样美好），此外还能了解用户与用户之间的协作、工作的流程。

这其实是一件很有意思的事情，你可以籍此离开计算机一段时间，观察一些对你而言比较陌生的事物，而它们通常都是五花八门的。你也许要去某所幼儿园，也许是某个大超市，说不定还可能是国家火箭航天科研生产中心。



很难明确地描述“现场研究”到底能够获得什么数据，这是由这个方法的本质决定的。它是一种体验，而不是某种具体的过程。你能观察和体验到的内容包括：

- 用户在工作的时候是否戴着耳机听音乐？
- 办公室里面是否吵闹？
- 当用户在操作出现问题后，是找人帮忙还是自己尝试着解决？
- 他们使用的鼠标有没有滚轴？
- 开会的时候是不是经常有人迟到或早退？参加会议的人是不是来去很自由？
- ……

你需要把这些“不太像数据”的信息记录下来，并且对其做出合理的解释，然后与设计团队的其他成员分享这些体验。你们的目的是分析这些信息对正在设计的产品会有什么影响，然后看看能不能改进用户的工作习惯。

有一家游戏设计公司开发了一套企业管理系统。这套系统集成了员工日常需要的绝大部分功能，非常实用。其中的内部邮件系统是一种方便公司上下沟通的机制。系统的开发人员为了方便用户，特地在首页放置了一个小版块以提醒用户目前有邮件需要确认。



但是开发人员没有注意到，公司里面有些部门是由从事图形设计方面工作的员工组成的，例如平面设计部门和动画部门。这些设计师普遍不喜欢受约束，厌倦公司的来往公告和通知，对那些枯燥的文字很反感。他们根本不屑花费时间去收邮件，也从来不发。他们认为有事拨电话就行，不需要什么麻烦的邮件。

所以，很多员工的内部邮箱塞满了未读邮件。他们对那个版块上日益增加的数字无动于衷。这样一来邮件系统对他们而言纯粹变成了摆设。

但是，并不能说这些邮件都是他们认为的“垃圾信件”。可能确实会有些工作流程需要他们进行 E-Mail 确认，也可能会有些涉及他们自身利益的通知。长期这样下去肯定会影响他们的工作。

这就是开发人员不太注重“现场研究”的结果。他们的想法很好，但是没有深入到用户身边去观察和体验。他们不知道还有这样一个“族群”，根本视内部邮件为无物。

也许我们可以针对这种情况提出一些解决建议。例如可以划分邮件的重要等级，当员工收到非常重要的邮件却又没有阅读的时候，系统在当天应不允许他们在系统中签退，或者再次强制提醒。



不看就别想下班

4.4.2　你是师傅，我是徒弟——“从师求教”过程中要注意的几个问题

“现场研究”强调的是到用户工作的环境中，在用户工作时观察，并和用户讨论他的工作。用户知道你（也就是研究人员）的存在，他们也知道自己是研究的一部分。你也会向用户提出各式各样相关的问题，这就好像木工师傅带徒弟一样，用户是师傅，而徒弟先在一旁站着看师傅干活，遇到不懂的就问。

在这个“从师求教”的过程中，你也要注意几个方面的问题。

不是在会客室

你是在用户正常使用产品的环境里观察他们，而不是去他们公司做客。在会客室里围着桌子谈话那不是观察，也不在现场。你要在他们工作的时候观察，在堆满了他们每天使用的产品的环境里向他们提问，这样才能够发现他们行为的所有重要细节。

如果你不太了解用户的思考过程，可以要求他“有声思考”，用语言描述自己的思维。你也可以在必要的时候向他们提问。当然，不要过于影响他们的正常工作。观察和讨论可以交替进行。

建立正确的关系

正确的关系就是师徒般的关系。很多时候用户总是没有摆正我们的位置，例如：

- **总把我们视为专家**

他们觉得“研究人员”都是很厉害的高级专家。想想，研究一个人和他的工作，一般人谁会想到做这些事情。但你必须要提醒他们，正是因为了解得不够才专门来研究。在这里，在他们的领域，他们自己才是专家，而你是去学习的。

- **总把我们当作采访者**

提问的过程总会让人感觉是在受采访。而事实也确实如此，一般都是我们提问题，他们回答。这种错觉导致的情况是，当你没有提问的时候他们就觉得你全都了解了，于是便被动地等待下一个问题。在这种时候你得强调，你需要他主动的指导，应该把你当成是刚来工作的新手。

- **总是以客相待**

还有些时候他们客气过头了。现场研究的过程中一直在给你端茶倒水，唯唯诺诺。但你是在工作，不是做客消遣。



该问就大胆问

如果在观察中你发现对某些问题有疑虑，必须马上提出来。一定要小心避免出现这种情况，就是不经过用户的验证，而自己片面地对事实做出解释或者假设。有时候你的理解并不能正确解释用户的行为。

所以，你需要让用户来解释这些问题。即使你能解释，也要把这些想法分享给用户，让他们来验证这些结论是否正确。如果你们已经确立了正确的师徒关系，用户就会很在意你的理解是否正确，而且会及时更正各种误解。

不要跑题太远

在“现场研究”中，用户是主角，所以他们往往会把问题引到自己感兴趣的地方。这没问题，因为你可以了解哪些问题是他们最关心的。

但是，你也不能让研究活动漫无边际地进行，不然会很浪费时间。你需要巧妙地引导他们，把研究的焦点集中在一定的主题上。

这就表示在进行研究之前，你就得准备一个大致的观察方向，并且保证按照这个方向进行下去。这并不是说要你准备好每一个问题，而是让你知道有哪些问题需要关注。

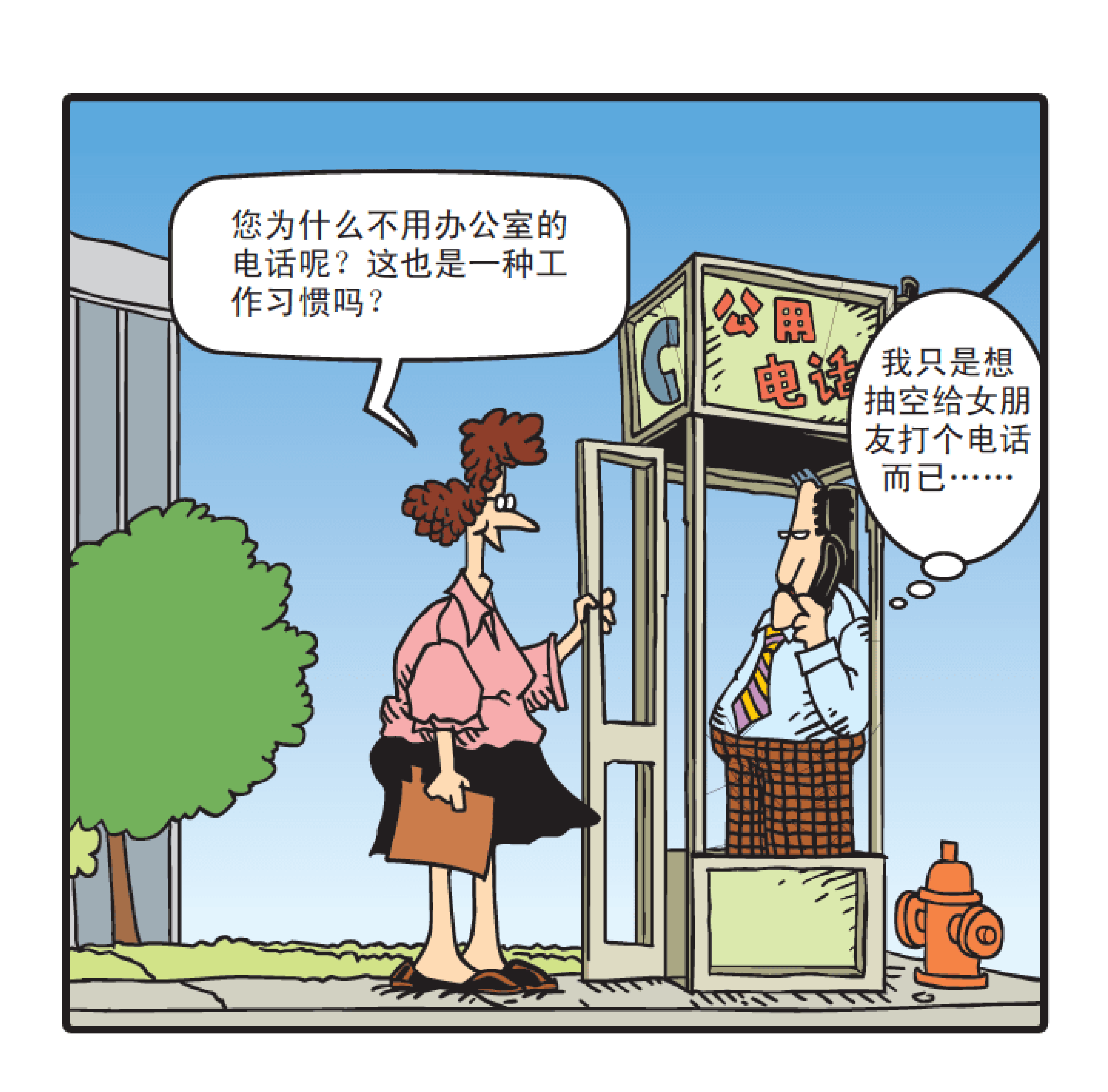
4.4.3　这不是你的地盘，别太不拘小节

在进行现场研究的时候，一定要注意合理性和礼节。这可是人家的地盘。贸然地插入到用户日常工作中去，也许会引起反感和不适："我知道这是为了能更好地支持我的工作，但是我每天已经够忙了，现在又来个家伙整天在我旁边呆着。"

所以，别一见面就进入正题。先培养一下积极、和谐的气氛，例如可以共进午餐，融洽气氛。在平时你们也可以谈论一些共同感兴趣的话题。表现出你的友善、轻松和易于接近也很必要，关键是让用户感觉不受拘束，轻松自在。

当然，不要迟到，不要为用户增添不必要的麻烦。例如：

- 提问时不要打断用户的任务或思考。
- 不要提带有倾向性或者误导性的问题。
- 尽量使用开放式的问题。
- 不要使用专业术语，他们不一定能理解。
- 不要帮助用户，你不是在为客户服务。帮助用户会让他们感到不好意思，那会影响研究效果。
- 摆正自己的位置，不要向用户卖弄你的产品……你是在做销售还是做研究？
- ……



由于现场研究是在用户的场所进行，所以你还需要注意一些其他细节：

- 在研究进行当中最好关掉手机（好吧如果你坚持，那就关掉手机的铃声）。
- 如果是在比较私人的场所，尽量避免使用用户的卫生间。
- 如果需要录像，事先征得用户及其上级的同意。
- 摄像机最好有足够的电池。除非用户同意，否则不要拔下它们的插头。
- ……

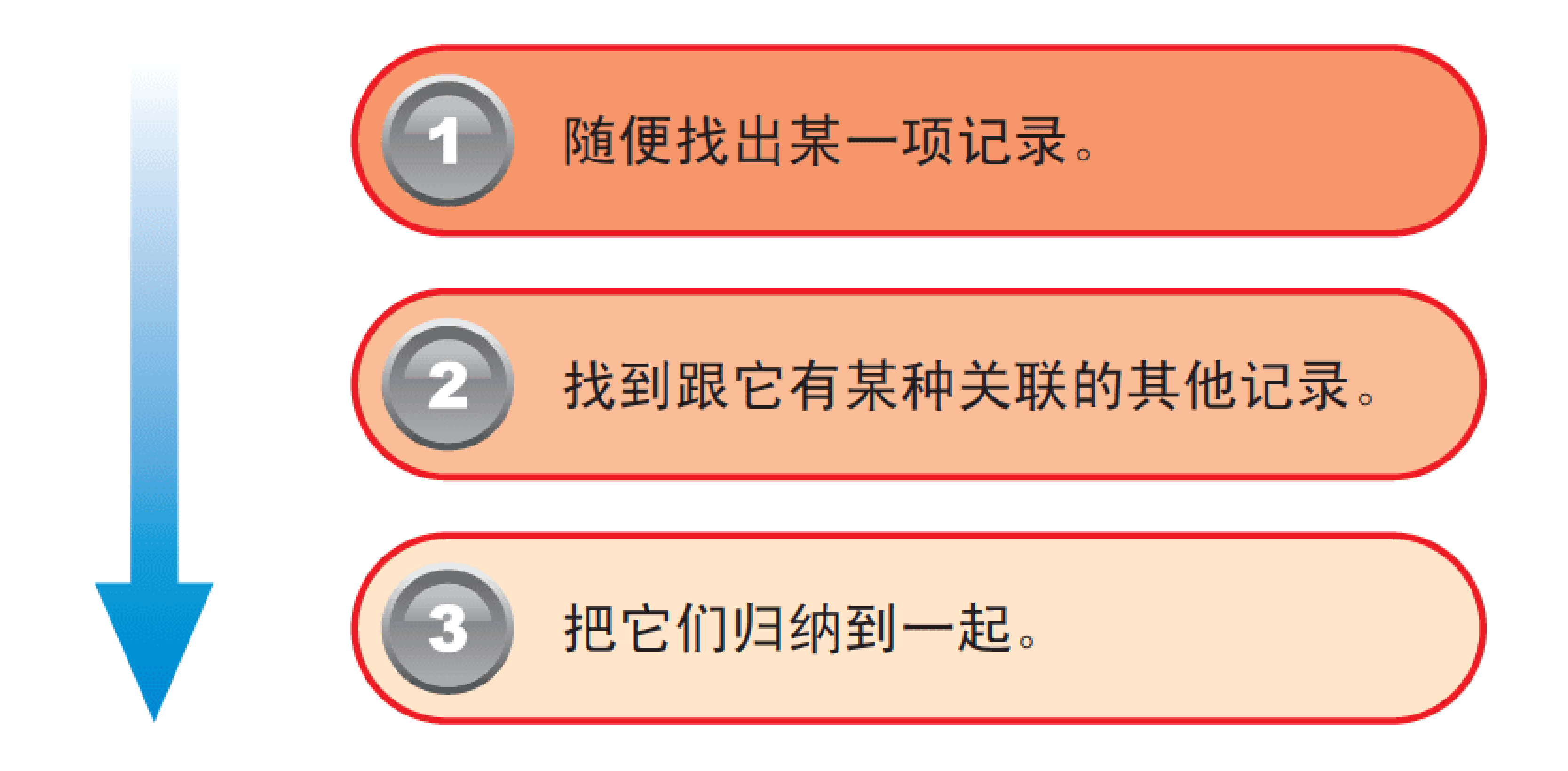
4.4.4 我们能得到什么？

“现场研究”后得到的信息很难被明确地描述出来。因为这些信息可能很大，而且很零散。但是你也不能拿着一堆资料却不加以整理。总不能像无头苍蝇一样找不到重点。

我们可以通过一种叫做“亲和图”的方式来把记录进行合并，整理为层次清晰的文档。

“亲和图”是 1953 年川喜田二郎（Jiro Kawakita）在尼泊尔探险时发明的。他用这种方法整理研究在野外的调查结果资料。这是一种把收集到的大量各种数据、资料，按照相互之间的亲和性进行归纳整理，使问题明朗化，从而获得统一认识的方法。

这是因为，不同的记录可能存在某种程度上的相似性，因此可以组合在一起。这是一种归纳的方法，即：



找出来的记录放在一边，然后继续这个过程，直到所有的记录都被分类。这个方法其实很简单，只要一点耐心谁都能做，也不需要任何计算。所谓的列出记录，就是把信息记在便条纸上，然后把它们都贴在墙上或者白板上进行整理。这样每个人都能很容易地看到，而且方便分类。

还是拿前面的那个内部邮件系统做例子。



写新邮件的界面

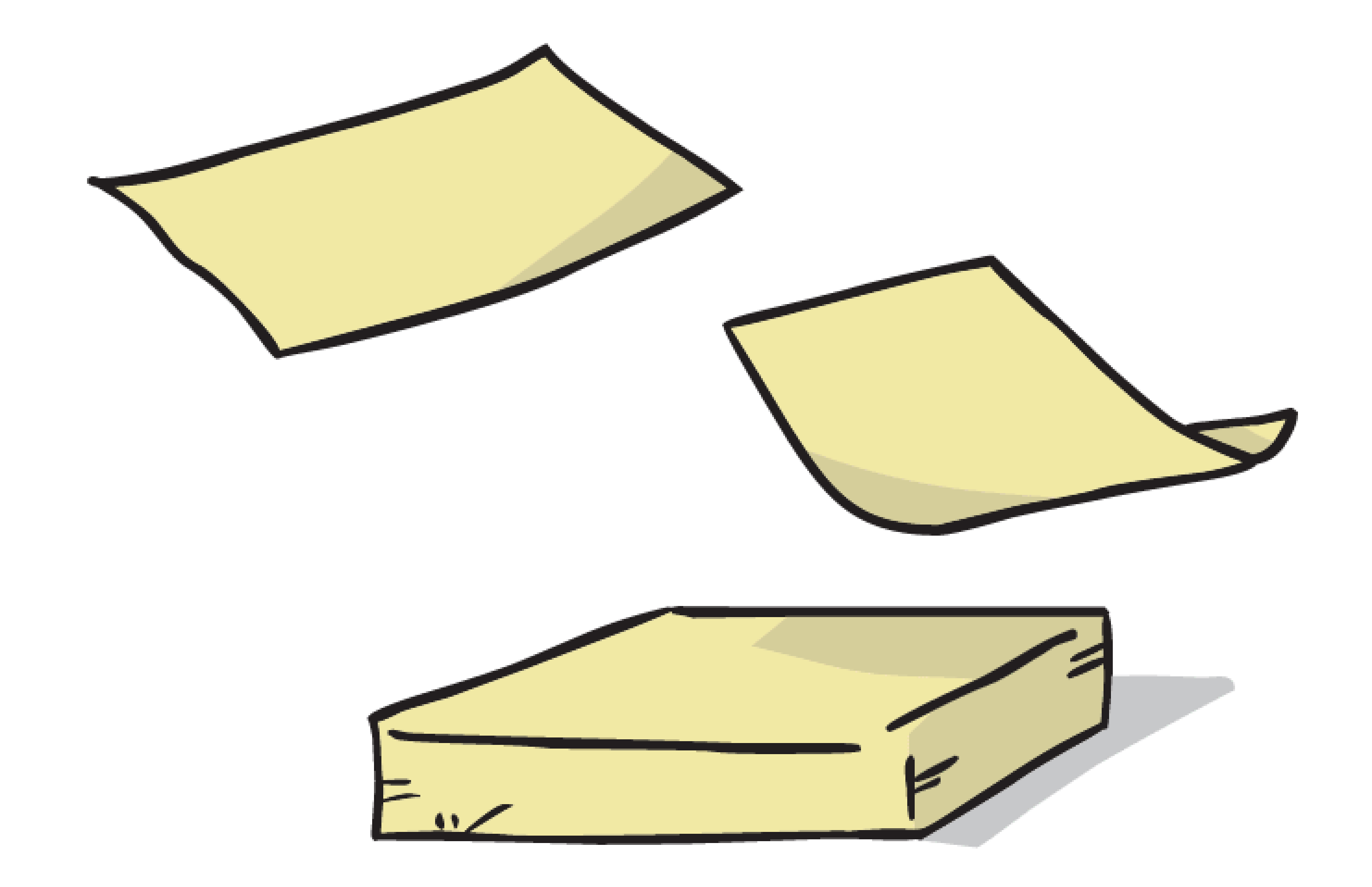
我们对这样一个内部邮件系统进行“现场研究”后会得到很多信息。不同的用户可能关注着不同的方面，也会有各种各样的看法，记录的时候肯定也是乱七八糟的。

这种时候你便可以用便条纸制作“亲和卡”，并且以颜色区分。注意，每项记录的文字应该简单、精炼、明了。最后的亲和图大概是下面这个样子（当然是一小部分）：



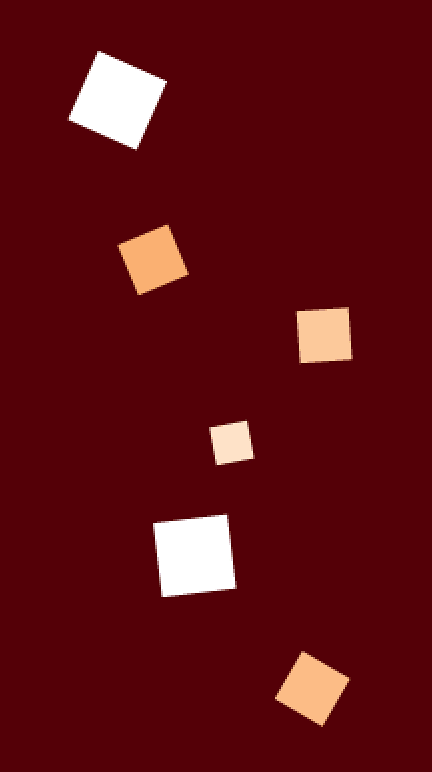
贴在白板上的亲和图

你看到的黄色便条纸是现场研究后获得的用户反馈意见。我根据某种逻辑关系把这些记录进行了分组，然后用绿色便条纸标明类别。这样就能知道用户对哪些部分都有哪些意见，一目了然。所以，在当我需要对某个部分进行改动或设计的时候，就能拿这些作为依据和参考。很明显这种表现方法比零散的资料要清晰得多。



第五章

文化的差异



一个老外对我说："你们中国太奇妙了，尤其是文字方面。比方说，'中国队大胜美国队'，是说中国队胜了；而'中国队大败美国队'，还是说中国队胜了。总之，胜利永远属于你们。"

众所周知，Web 是开放性的。不管我是哪个国家、哪个地区的人，只要我能上网，那么就可能会浏览到你的网站，成为你的用户。也许我看不懂页面上的文字，但是也仍然可能对某些东西感兴趣。没错，每个产品都会有自己面向的特定用户，但任何人都不会说："老外来这个网站没意义，我不欢迎他们。"

同时，国际间的技术合作和交流也越来越频繁。某个国家的设计人员为其他国家的客户开发网站或产品早已不足为奇（我曾经参与过一个游戏的设计，它有 11 种语言）。如果你只以自己熟悉的思维方式来考虑用户界面和设计，就无法保证每个用户都能顺利地理解。

这就好像我们国家的麦当劳和美国的麦当劳并不相同一样。他们考虑了中国人的口味，所以大部分食品都比那边更甜一些。

如果你想开发的是像"Google 文件"那样针对全球用户的产品，那么就更应当注意不同国家和地区之间的文化背景差异。



5.1 什么是文化的差异？

别以为文化的差异只是语言和文字的不同。

在不同的文化背景下，人们的思考模式、决策、情感偏好以及解决问题的方法等等都存在着相

当大的差别。众所周知，美国人自由开放，而中国人则内向腼腆。这种差别是千百年来不同的历史、风俗、传统教育等造成的，是一时之间没法扭转的事实。

与此同时，就算是在使用同样语言文字的同一个群体中，也存在着文化差异。例如在中国总有人说北方人粗犷豪迈，而南方人精打细算。美国也有所谓东海岸和西海岸之间的差异。

但反过来说，即使是使用不同语言文字的群体，也并不代表在文化上就没有共通点。例如对中国人来说，欧洲人和美国人的思维方式就存在许多共通点（都是“西方人”），而对欧美来说，中国人、韩国人和日本人也比较相似。

在那些经济快速发展、社会高速变化的国家和地区（例如中国），大众文化也在不停地发生变化。例如两代人之间的代沟非常明显。中国的年轻人更容易也更热衷于接受西方文化。

有些人做过很多文化差异的研究，其中 Geert Hofstede（荷兰 Maastricht 大学组织机构的人类学和国际管理名誉教授）研究了 53 个国家和地区的文化差异后，提出了“五个文化差异维度”这一概念。这也许是最广为人知的文化差异模型：

- **权力距离**

  组织和团体（例如国家和家庭）中，弱势群体接受与期待权力不平等分配的程度。

- **个人主义和集体主义**

  表示个人与群体的联系程度。“集体主义”这个词在这里指的是群体，而不是政府。

- **男性特质和女性特质**

  指性别间的角色分布。“男性特质”代表着武断和竞争性，“女性特质”则更具有温和、关怀的价值标准。

- **不确定性规避**

  这是衡量一个社会对于不确定与疑问的容忍程度。“不确定性规避文化”的国家里的人们更情绪化，为内部的情感力量所驱动。而相反的“不确定性接受文化”则更容忍与常规不同的选择。他们更冷静、更富于思考，不会因为环境而表达情感。

- **长期取向与短期取向**

  这是他根据在中国的研究成果增加的。长期取向的价值观注重节约与坚定，短期取向的价值观则尊重传统，履行社会责任，而且很爱“面子”。

让我们具体看看中国与美国人的差异大概是个什么样子：

- 美国人不喜欢高度结构化的信息，他们偏好散漫、人性化。而中国人则相反。而且美国人

不太重视权威、证书、公章等等这些概念。但对中国人来说，有时候老总的相片会产生很大的影响。

- 美国人更自我，强调事实，喜欢与众不同，讲究创新和个人英雄主义（看看那些好莱坞大片）。而中国人则一直在避免从群体中孤立，而且强调与他人的关联（“团结就是力量”）。
- 美国人之间性别的角色比较模糊，还有相当一部分的女权主义者；中国男女性别的区分则更为明显（日本更为强烈）。
- 美国人的思维跳跃性比较强，因此需要避免容易迷失的设计。他们表面化的思维方式决定了页面上不能有太多的选择，提供的数据也必须有限，而且最好是能够预测结果的操作。导航对他们来说更为重要。中国用户则更为习惯打开新窗口浏览。他们上网时可以接受更多的风险，喜欢页面中包括更大的信息量。
- 美国人更注重事实以及确定性，对目标的结果有实时的期望。中国人则重视实用价值，对结果更有耐心。另外，美国人对信息的可信度是以规则为标准的，中国用户更多注重关系。

但就像我之前所说的，中国的文化一直在发生着变化，我们也需要对其进行不断地认知，从而不断地修正。

5.2 对于Web和Web-based产品来说

除了上面说的一些差别之外，不同文化的用户对于时间取向、认知程度、网页视觉反映等等方面都存在着差异。

5.2.1 一次要做几件事？——不同文化的用户对“时间”和“效率”的不同认识

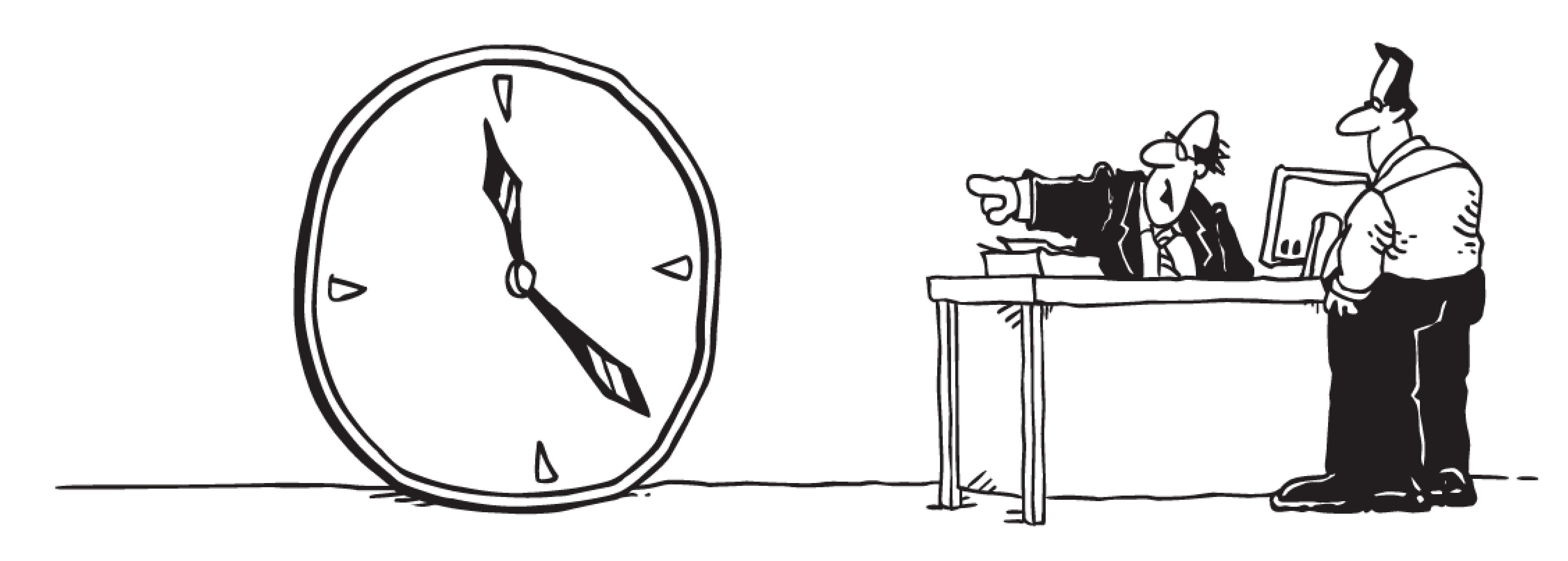
有一些专家认为，不同文化的用户对如何看待“时间”和“效率”有着明显的不同。例如是否遵守时间、如何安排工作和生活，以及如何调整时间进度等等。这些因素会影响对网页和其他交互产品的操作方式。

这些专家通过观察认为，德国、英国、荷兰、芬兰、美国以及澳大利亚的人比较倾向于一次完成一个任务。这些国家的人把时间视为线性的，分为许多段落，一次专注于一件事情，而且比较遵守规则与程序。他们不习惯同时做许多事情。这些用户在上网的时候也表现出线性的浏览模式，他们点选更多的链接，看完一个页面就关闭一个，所以花的时间也比较长。

而意大利、法国、西班牙、巴西和印度的人，则更喜欢一次做多件事。他们更为弹性地运用时间，

统筹安排，不一定严格遵守规则、流程与时间表。所以他们偏好同时进行好几项任务，交替地运用时间。例如边听音乐边聊天，用电脑写报告，同时还在下载新游戏。

中国用户更喜欢一次做一件事，但是大多数情况下却被迫必须同时进行多项任务。例如使用搜索引擎的时候，他们可能一次就点开很多搜索结果链接。如果某些页面打开的速度比较慢，他们会先放在一边，然后继续搜索结果，等到这些页面打开后再进行浏览。这有时候会让人感觉有点疲劳。



这样能够方便员工看时间

- **一次做一件事的用户**

如果用户在一段时间内只专注一个任务，可能会导致忽视整体的情况，甚至会忽略对警告的反应。而且这些用户因为过于重视清楚的流程，如果系统出现不明情况或发生变化的时候，他们一般不会根据新的情况来做出相应的改变。

对于这种用户来说，你需要提供清晰的结构和流程，而且尽量做到面面俱到。提供减少失误的辅助系统，当出现了错误或意外情况时，最好能提供完善的提示和说明。

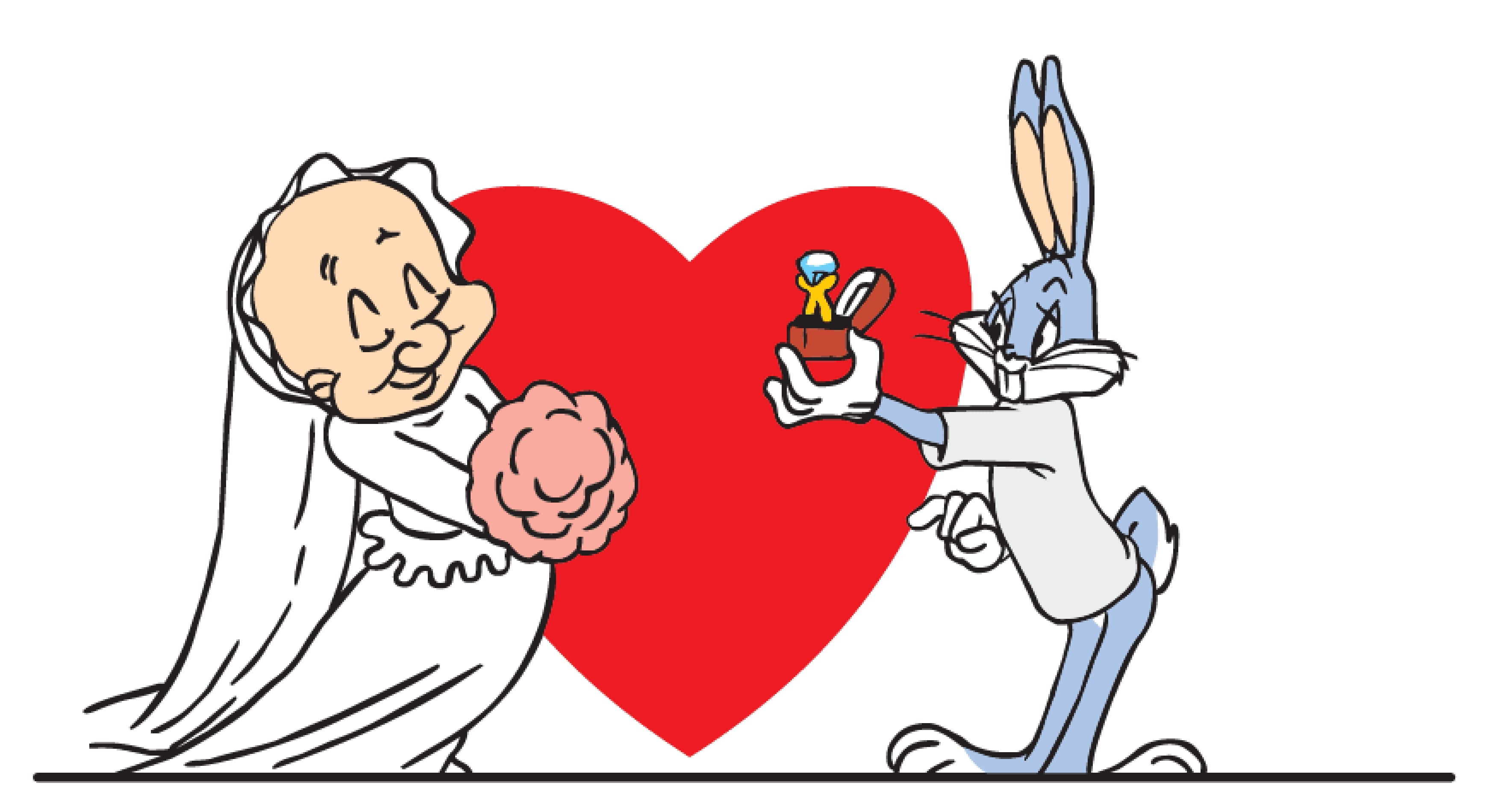
- **一次做好几件事的用户**

这些人在运用时间方面更有弹性，但同时也充满了不确定。他们可能会在网页或者系统之间转来转去，希望全面掌握整体的情况，但也可能因为这些转换而产生失误。例如他们可能因为忘记之前的状况而选择错误的步骤或者规则，或者把这个网站的某些特性记成另外一个网站。

对于这种用户来说，必要的提示和引导是非常重要的。任务与任务之间应有所区别，别让他们迷失了方向。明确告诉他们，我的特性是我所独有的，别记混了。如果是非常重要的操作，禁止他们切换到其他任务去也是个好办法。

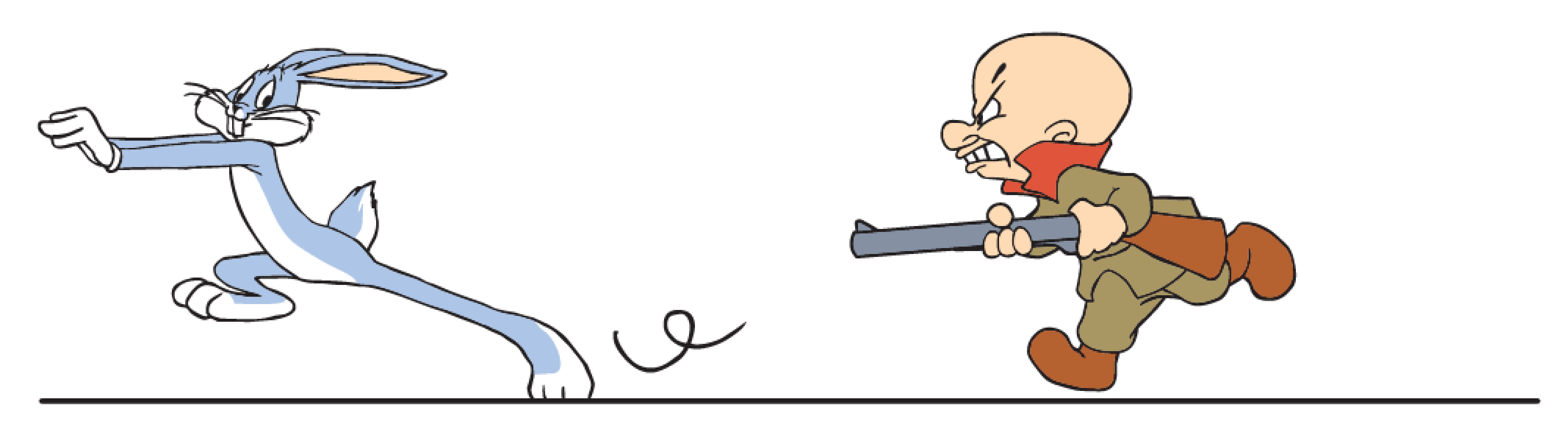
5.2.2 高度认知和低度认知

有些人对于大部分该具备的信息都已经了然于心，他们在交互时只需要了解很少的信息就能正常完成任务。想想两个一起长大的双胞胎，他们彼此之间可能一个眼神就能知道对方想表达的意思，沟通非常简短而有效率。



嗯，高度认知

低度认知的人则正好相反，他们需要阅读大量的信息，必要的情况下可能还要一些符号或者图案加以辅助。如果没有足够的信息量，他们很容易感到怀疑、迷惘和不明所以。这不是说他们都是"笨蛋"，只是因为他们的思维更注重细节，需要证实再证实才可能去认同一件事情。



……我想他们彼此现在属于低度认知

很多专家认为中国是高度认知的国家之一。中国人是高度认知人群中的最高者，这是值得骄傲的一件事情。高度认知的人偏好自由与广泛的信息流通，不管是搜集信息还是处理或传递信息，彼此之间的寻求和给予都非常频繁，属于高度的信息共享（一个不太正面的例子：想想我们是如何传播和下载电影、音乐和软件的）。

而在德国等低度认知的区域，信息相当的零散和片面，信息通常也不会共享，而且通常都经过筛选。这些地区的人在处理事物时，总是习惯要对该事物有全盘的了解才肯信任它。

如果你为高度认知的用户群提供过多的信息，他们可能会感到不耐烦，或者感觉被人“瞧不起”。而当你没有提供足够的信息量给那些低度认知的用户时，他们就会感到困惑，似乎遗漏了重要的线索。

对于 Web 和 Web-based 产品而言，高度认知的用户浏览网络的速度比较快。他们往往比低度认知的用户要更快捷地到达某个页面，因为他们所点击的超链接比较少。这些人更会找“捷径”，一箭中的，对于不相关的内容就自然而然地忽略。但正是由于这种习惯，他们不愿意去对信息进行全盘的了解（因为这很麻烦），所以更容易产生“迷失”的现象，在网络的虚拟空间中失去方向感和位置感。他们可能几步操作就来到了“某处”，但是却不知道该如何继续“前进”，因为他们没有去了解结构和规则。

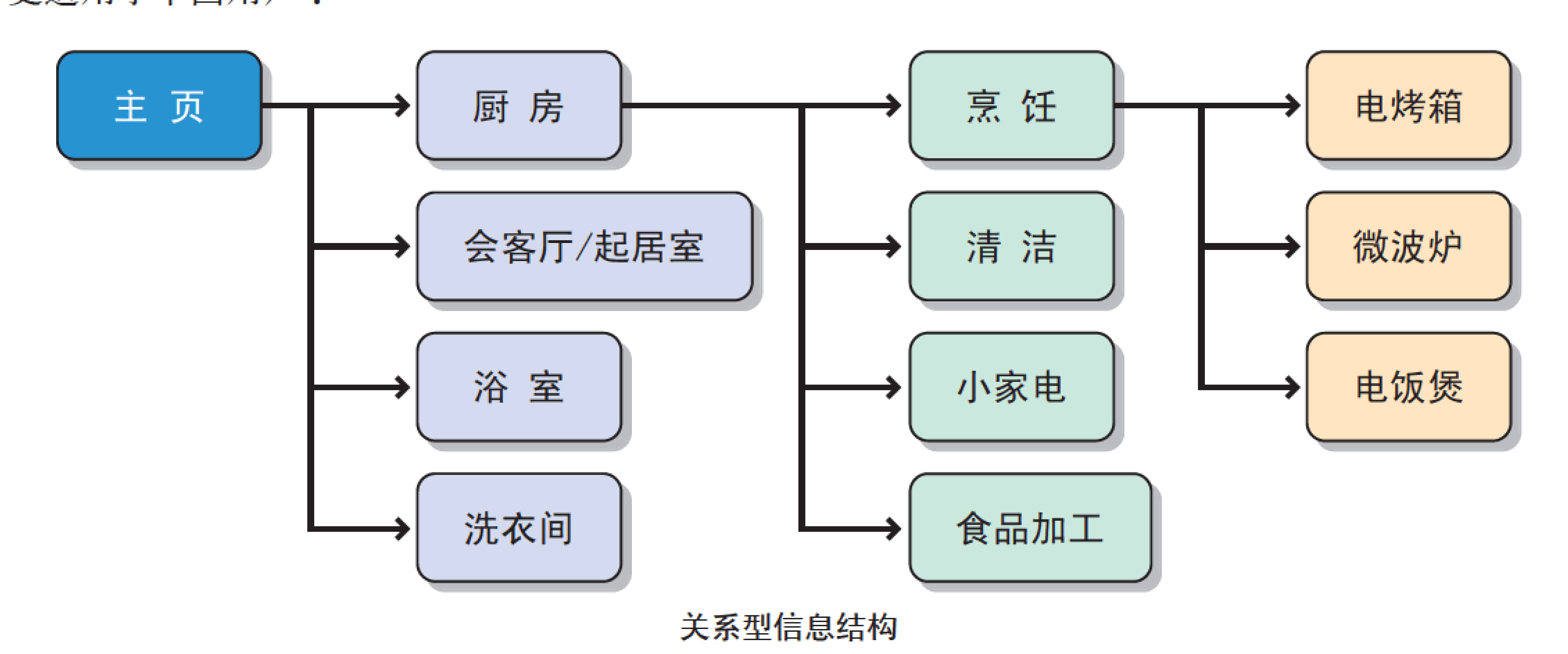
相对而言，低度认知的用户虽然浏览的速度比较慢，打开的网页比较多，但却不容易在上网时感到迷失方向，而且也不会失去目标。

如果你的目标用户较为偏向高度认知，就应该在每个网页或界面上都完整详细地呈现出信息架构，提供较多的导航支持，从而避免他们迷失其中。

5.2.3　功能还是关系？——中国用户和美国用户的区别

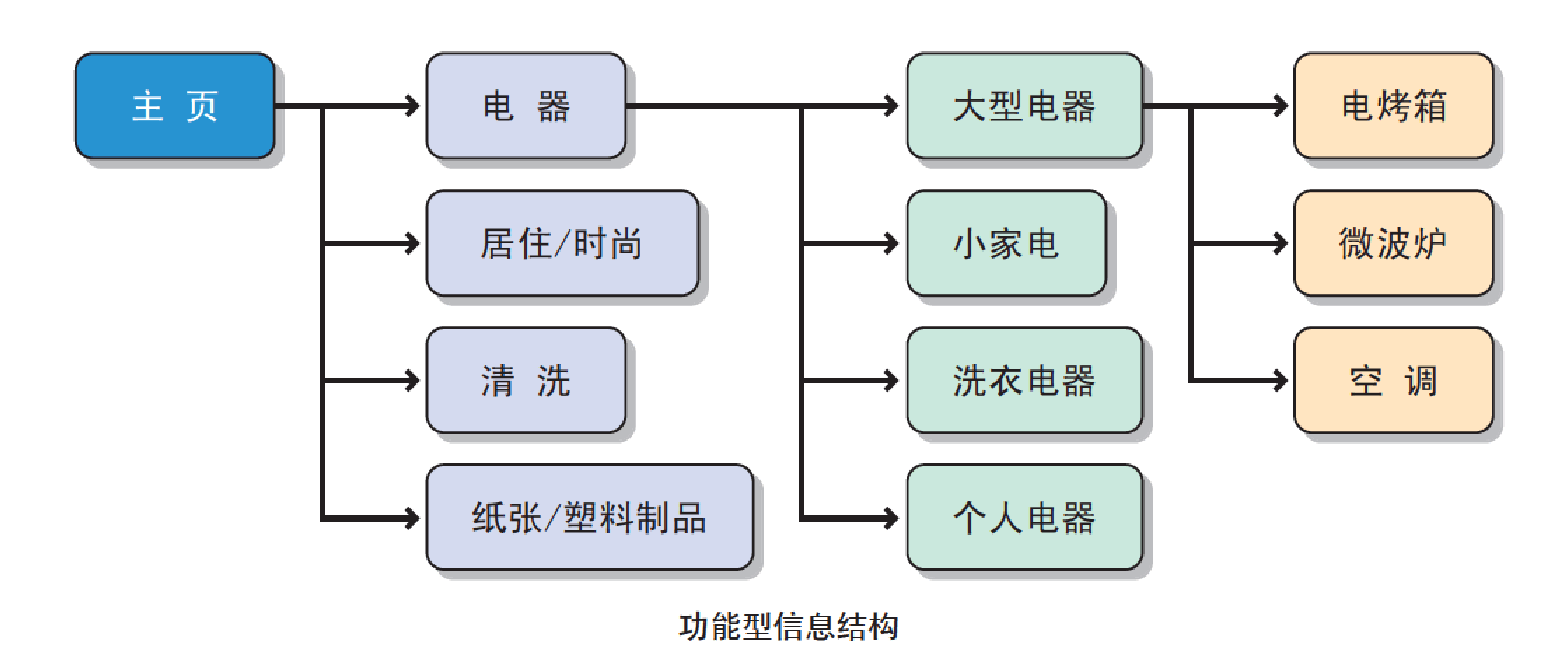
这个问题涉及到 Web 或 Web-based 产品的信息架构设计。有些研究表明，中国用户使用关系型具体分类系统更加有效，而美国用户则更偏向于使用功能型抽象分类系统。

所谓“关系型”分类系统，体现出实际生活中事物之间的联系，是一种比较具体、表面化的分类方式。如果有一个在线超市的网站，按下列方式进行分类就属于“关系型”分类系统，而且更适用于中国用户：



关系型信息结构

而“功能型”分类系统则更偏向于体现实际功能之间的差异，这种分类方式更为抽象一些。同样是在线超市的例子，下列分类方式则更适合于美国用户：



功能型信息结构

很明显这种导航分类在我们中国人看来是不习惯的。

5.2.4 网页和界面的视觉反映——中国用户的神奇能力

有一个非常有意思的情况是：中国的门户网站与国外相比，信息量大得令人吃惊。而且它们的主页非常长，普遍的情况下需要翻屏5到6次才能浏览完整个页面。这些门户网站都试图在超长的主页上提供所有可使用的信息，而且充斥着各种各样的链接、不同大小的文字、争奇斗艳的色彩，以及闪耀华丽的动画广告。甚至于满页面漂浮的动画也非常普遍，像搜狐这样的大型门户网站还有整屏的背投广告。



不管你相信与否，这就是新浪的首页



搜狐的背投广告，每当打开搜狐网站后就覆盖了整个屏幕

这些设计在几乎所有国际上的人机交互研究中，都被认为存在严重的可用性问题，是绝对不可行的设计。因为用户在超量信息的网页上很容易忽略所要找的链接。但是在我们国家，门户网站显示这种“海量”的信息却是普遍的标准。再来看看 Yahoo! ,同样是门户网站(还是国际性大门户)，首页也只有这么多内容，而且所有的广告都是静态的，绝不会干扰用户的注意力。



Yahoo! 的首页

是中国的用户与众不同吗？还是说中国的用户具有某种能力，根本不把超量的信息当成问题？也许长时间以来他们已经习惯了这种方式，从而培养出不同的偏好，或者说锻炼出了不同的能力。

另一个常见的问题是中国门户网站上任何页面都泛滥成灾的动画广告，这同样也是不被国际交互领域所认同的设计，因为它们极大地干扰了用户浏览正常信息。有时候一些网站甚至在网页

正文的部分插入大面积的动画广告：



有时候希望在广告的位置上贴块胶布

有人研究得出的结论是，中国用户能够察觉到动画闪现和移动的规律，从而在脑中能忽视动画的干扰，做到视而不见。但我认为并不是所有人都有能力做到这一点，因为仍然有很多用户在自己所用的 Web 浏览器中屏蔽了 Flash 插件，干脆从此不看任何 Flash 动画，以此达到眼前“清净”的目的。

具有中国特色的网站设计对中国的用户来说，存在的问题很多，这和国际上交互领域所得出的结论是一样的。但是，可能中国用户对此已经形成了特定的使用习惯，所以不觉得问题非常严重。

5.2.5 谁的错？——有人批评产品有人自我批评

不同文化背景下的用户，在 Web 或 Web-based 产品的使用中出现问题或者效率低下的时候，对于问题产生原因的认识也不一致。也就是说，哪怕是同样的产品，在不同文化下用户的满意度也不一样。

Yahoo! 曾经做过一项研究，邀请中国台湾和芝加哥的用户分别在他们的网站上执行同样的搜索任务。结果发现，两组用户对搜索的结果和效率相差不大，但是对结果的满意程度却有明显的差别。

芝加哥的用户效率不高，他们提出了一些产品上存在的问题。而台湾地区的用户也对结果不满意，但是却更趋向于自我批评。他们认为效率不高是自身的原因，因为他们觉得自己还不太了解 Yahoo! 的搜索引擎，也不知道该搜些什么。同时，他们认为可以经过强化练习来提高效率。

中国包括其他亚洲地区的用户，当感觉到出现问题的时候，更多地把原因归结到自己身上，而不是认为产品需要改善。这可能有一部分对自我信心不足的原因。这种倾向一方面表示亚洲地区的用户更善于提高对自己的要求，他们很可能在初次接触产品时更愿意先自我学习，而不是挑毛病；但另一方面，他们较低的自我评价也不利于对产品的改善。例如在对产品进行用户测试的时候，他们很可能不善于对产品提出有建设性的要求。

5.3 国际化和本土化设计中要注意的问题

随着中国和世界日益广泛的交流，很多在世界各地开发的产品逐步进入中国市场，而同时在中国设计的产品也会被输出到国外。

这对于你来说，就是产品国际化和本土化的问题。

- **“国际化”**是指设计的产品需要尽量适应绝大部分国家或地区。也就是产品的“兼容性”。
- **“本土化”**则是指针对某个国家或地区设计的产品，需要专门进行某些调整。也就是产品的“针对性”。

这两者相互之间并不矛盾。对我们来说，产品的国际化是一种普遍意义上的追求，开发的网站或产品应当在一定程度上满足世界各地用户的可用性需求。但是，如果某个重点地区的用户在使用习惯上与其他地区有相当大的差别，你就需要进行调整以适应这些用户。

例如，一个公司的网站有中文和英文两个版本，这就是“讲究国际化”的设计；但如果这个公司在德国有很多重要的客户，那么他们就可能需要专门再开发一个德语的版本，这就是网站“本土化”的策略。

下面是我们在国际化或本土化设计中所需要注意的一些方面：

注意适配分辨率的大小

在我们国家，1024×768 像素分辨率的显示器是主流，有些幸运的设计师使用的还是 1280×

1024 的分辨率（比如说我），更别提 1600×1200 的庞然大物了。但是不一定每个国家都有像我们这样好的计算机硬件环境。很多非洲国家通用的仍旧是 800×600 像素的分辨率。此外，你还需要注意的是现在宽屏笔记本电脑的流行，屏幕更宽也就意味着高度更小，一屏内显示的内容反而可能会有所减少（比如常见的 1280×800 像素分辨率）。



1280×1024分辨率，一切都很美好



可是1280×800分辨率下就不是那么完美无“缺”了

尽量多用被广泛接受的图标

在不会太影响（尤其是网站的）页面体积的情况下，你应该多使用一些不带有地区性限制的图标。例如，使用房屋住宅图案表示“首页”或“起始点”，不管在世界上哪个国家或地区都是很直观的。如果用户看不懂与之相连的文字，或者视力不太好，他们同样也可以通过这种图标来理解其含义。而且这也能帮助他们减少记忆量。注意，图标一定不要表意不明。



绘制图标时注意地域性

由于不同地区有不同的环境，所以表现同样内容的标志也有可能不相同。伊斯兰教国家的住宅图形可能会被其他国家的人当作“宫殿”或“主题公园”图标。再比如“邮寄”或“电子邮件”图标，通常我们使用邮箱或者邮筒的图案，但是世界各地邮箱或邮筒的形状各不相同。很可能某个地区的人们能立即辨识出来的邮箱图形，另外一个地区的人们却从来没有见过。这种情况下，用邮戳或者邮票的形式就能解决这个问题，因为邮票是世界通用的。



翻译时使用用户习惯的表达方式

每种语言都有自己的成语或俗语，如果使用得当，就会有非常好的效果。例如英语原文是“Remember the past and it will guide your future.”，直译为“记住过去，将会引导我们的将来”当然也算不错，但如果改成“前事不忘后事之师”就会更“本土化”一些。还有些俗语弄错了意思就会闹笑话。“be all ears”的意思是洗耳恭听，这个还好理解，“rat race”呢？老鼠赛跑？

注意不同地区使用的单位和格式

这个很容易理解。我的一个英国朋友在向中国人描述尺寸大小的时候，总要事先在脑子里把英尺换算为公尺单位。因为大部分中国人对英寸英尺并没有直观的概念。

注意文字的输入

我们主要需要注意一下全角和半角的输入模式。计算机系统中，中文字是以双字节方式存储的，而标准英文字母则是单字节。如果在中文全角模式中输入英文字母、标点或数字，也同样会以双字节方式显示。比如经常有人把“It’s”输入成“It’s”，其中的引号是不规范的。另外，如果用户以全角模式输入电话号码或者密码，系统就可能会错误地处理这些数据。

1 2 3 4 5 6 7 8 9 0 a b c A B C , . ; : " "

１２３４５６７８９０ａｂｃＡＢＣ，。；：“ ”

上面是半角输入，下面是全角输入

避免出现对某些地区不适用的信息

如果只是针对部分地区推出的产品或服务，就不要在全球范围内打广告。你不会在 Apple 中国地区的网站上找到任何关于 iPhone 的信息。如果你没有专门针对各个地区的网站版本，那么至少请在广告上注明该活动针对的范围。另外，在网页上提供有关经济、政治观点和地区间关系的内容时，更需要格外注意，否则很可能极大伤害用户的情感甚至引起法律纠纷。

其他方面

此外，你还需要充分考虑不同地区用户与用户之间不同的道德观、价值观、沟通方式以及特殊的喜恶等方面，这方面的一些例子包括：

- **手势和体态的意义**

  很多手势或者动作在不同地区所表达的意义完全不同。比如在有些国家（例如阿尔巴尼亚、印度和巴基斯坦）人们摇头表示同意，点头却表示不同意。

- **人种、性别、宗教信仰**

设计图形标志的时候一定要注意。老外也不一定全部都信耶稣。

- **喜好和习惯**

例如大多数中国人认为 8 是吉祥数字，但西方国家则更多认为 7 是吉祥数字。

- **交往和相处**

礼仪、称呼和用语各个地区都不一样。有些国家直接用人的名字来称呼对方，可以表示亲近；但另一些国家在不是很熟悉的情况下只能以姓氏来称呼对方，否则就会被视为冒犯和不识体统。

- **颜色的使用**

有些国家把高醒目的红色标识为警告色，表示“禁止”，但在中国红色同时却代表着喜庆和吉祥。与此同时，在医院等场所警告色又变成了黄色，这是因为红色和血液的颜色太相近容易混淆。

很明显，人类文化差异的复杂性远远不止上述的这些因素。你在设计中可能会遇到各种各样的问题，而解决它们需要广泛的知识背景和经验，而通常这都是超出一般设计范畴之外的。与此同时，它们也意味着你需要更多的开发时间和资源。

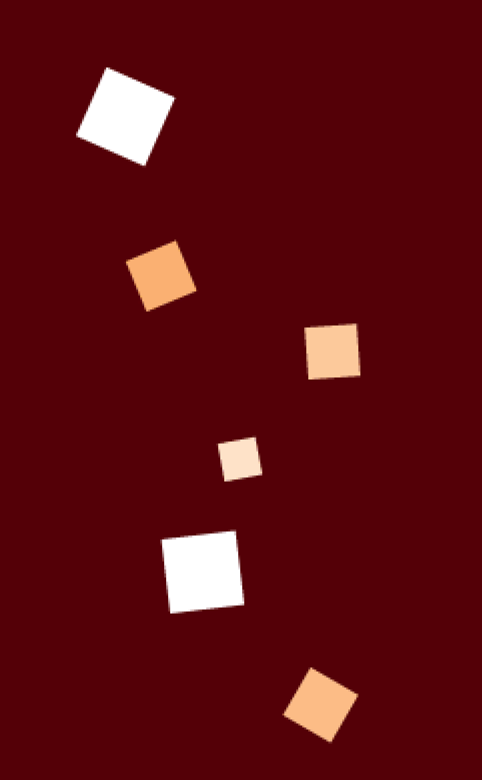
但不要因此而忽视这些因素。如果你开发的产品将在不同的文化环境中使用，或者说产品有相当数量的用户来自于不同的文化环境，忽视文化的差异很可能会导致很多可用性方面的问题，甚至可能是整个产品的失败，而这是非常危险的。这些问题越早被关注，就会解决得越好，最终的成本也会越低。

所以你需要在设计过程中做好以下方面的工作：

- 在项目开始阶段，就全面完整地定义产品所面向的所有文化环境和用户习惯。
- 在了解用户及需求的阶段，就要针对不同文化背景的用户，同时也需要邀请他们参与测试。
- 在设计过程中，时时注意顾及全部的用户，注重“全球化”的设计主张。
- 有条件的情况下，邀请具有国际化和本土化设计经验的专业人员，对各个环节提供指导和咨询。

第六章

网页的可用性



既然他们都视而不见，这么做还有意义吗？我还不如把时间用来做点公益劳动，起码那样还有人看得到我的付出。

我们一直都在说可用性的问题。强调“以用户为中心”的原则就是为了充分地理解用户和他们的任务，从而最大限度地提升产品的可用性，让他们能够“无需思考”地使用产品，从而把注意力集中在任务本身。

这也就是说，不管是 Web 也好，Web-based 产品也好，用户界面不应给他们带来不必要的困扰。用户访问页面或者使用产品，他们的目的不是为了“与产品发生交互”本身，而是要完成某项任务。

有一句话叫做“最好的界面是让用户感觉不到界面”。它的意思是说，好的用户界面应该具有最高的可用性。这种良好的可用性能帮助用户把注意力集中在任务上面，而不是被迫去思考自己的方法是不是合乎规范，会不会带来不好的结果。如果用户在使用网页或产品的过程中感到举棋不定、畏首畏尾和心烦意乱，那只能说明这是一款失败的设计。

我在这里把“可用性”归纳为以下的十六字方针 ：

不言而喻　简单易学　轻松自如　没有风险

在这一章和下面一章，我将分别从网页和 Web-based 产品两个方面来尝试解答下面的问题 ：

1) **可用性有多重要?**

2) **如何提升可用性?**

6.1 人们是如何使用网页的?

人们在上网的时候一般是什么样的?

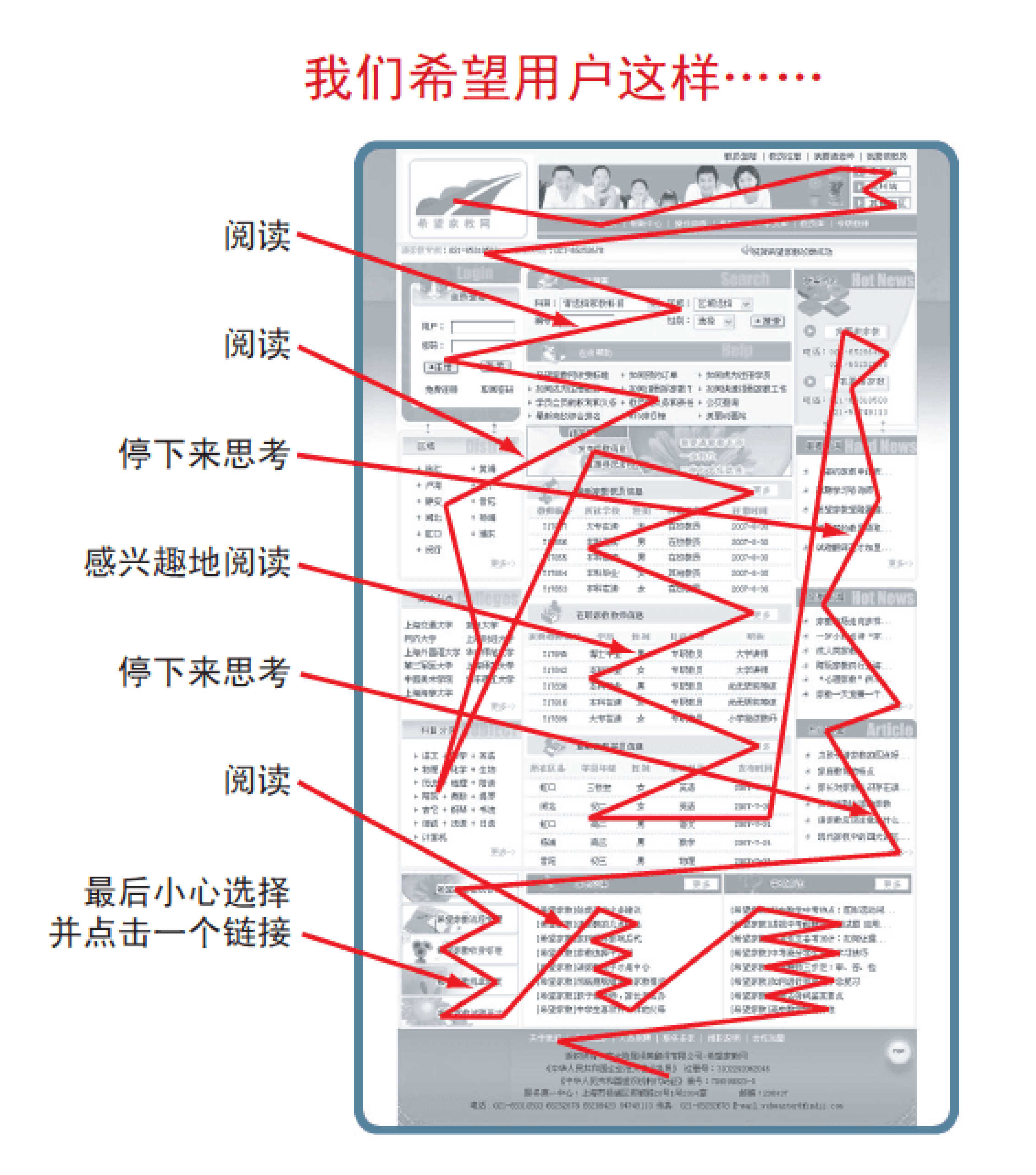
他们实际使用网络的方式，完全不是你想象的那样。尤其是当你正在设计网页的时候。

当我们创建网站的时候，我们认为用户打开网页之后都会仔细阅读每一个文字，看清楚整个网页的结构，思考每一张图片所蕴含的意义，然后经过仔细的权衡，小心翼翼地点击某个超链接或按钮。

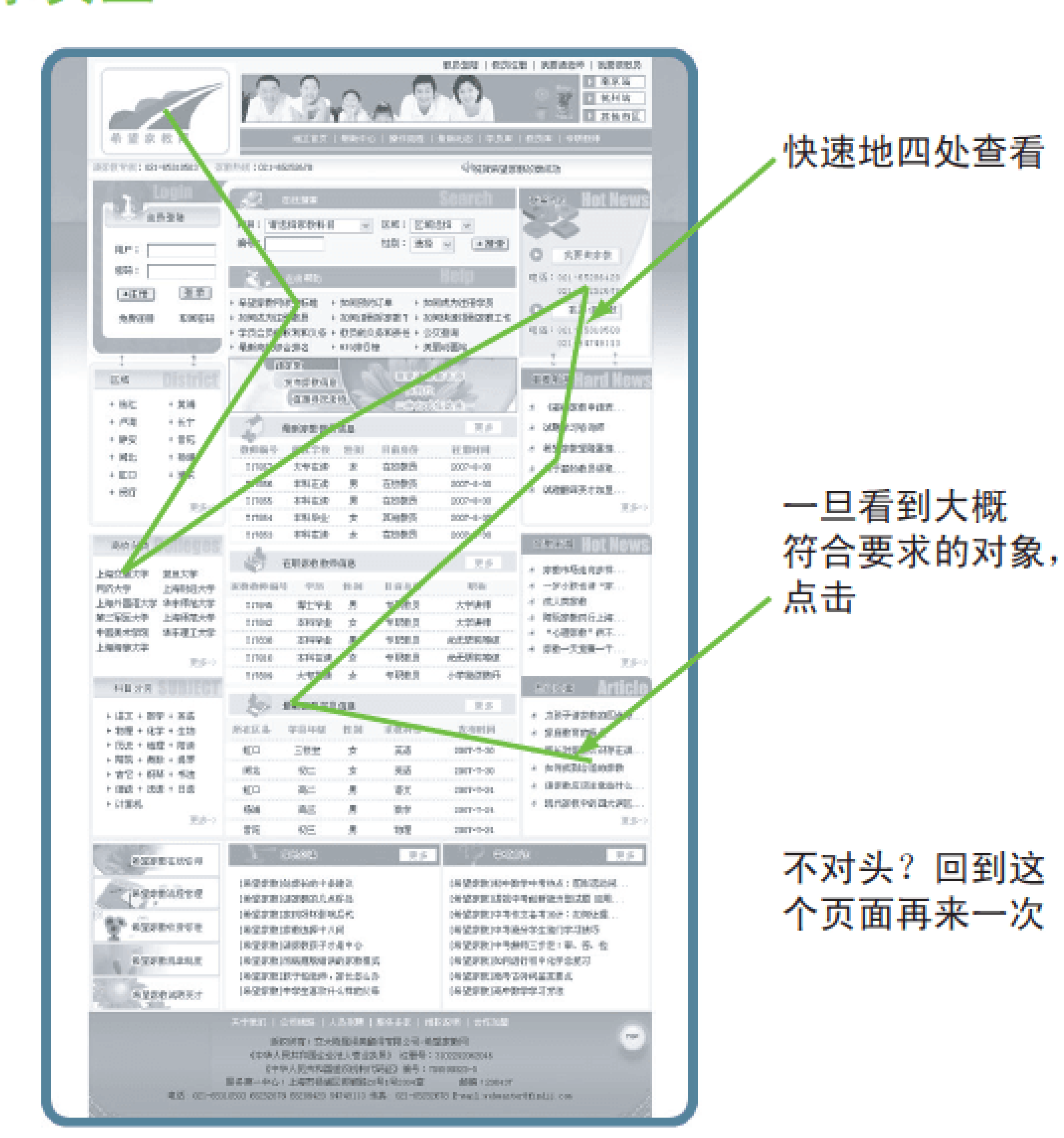
但实际上用户们使用网站的方式和我们的想象之间有很大差距。他们只是在每个页面上瞥一眼，眼光无顺序地扫过一些文字，然后点击第一个令他们感兴趣的或者大致符合他们寻找目标的链接。通常来说页面上的很多部分他们看都不看，也不感兴趣。

我并不是说用户都是些注意力不集中而且毫无理性的家伙。在他们的头脑中，只是关注着他

们来到这个网页所要达到的目标。比如说想买一款新的手机，或者找点工作所需要的资料。即使在他们无聊的时候，他们光临网站也只是为了找点能够解闷的东西，欣赏网站本身并不是他们的目的（当然学习如何设计网站的时候除外）。



事实上……



但这并不是说你就可以放弃细节、随意设计。清晰的结构、美观的图片和没有歧义的文字都能帮助用户快速地找到自己感兴趣的东西。如果用户稍不顺心或者不耐烦，他们很可能马上就关闭这个网页，拂袖而去。有时候他们就像一些脾气不好的顾客，对待他们稍有不慎就再也无法挽回。

6.1.1 他们很会节省时间

在大多数时间里，人们只会花很少的时间来阅读大部分的页面。他们所做的仅仅只是匆忙地大致浏览（或者说“扫描”更合适），看看有什么文字或图片能够吸引他们的注意力。他们处于一种“高速公路”的状态，路边的景色呼啸而过。

只有当来到“最终”的页面时他们才会把 110 公里的时速降到 12 公里以下。因为这里是他们的目的地：某篇体育新闻、某份技术文档或者某个问题的答案。这里才是他们真正想要驻足观光的地方。至于那些其他的页面……就好比你每天都乘坐地铁，但并不一定清楚地铁站里面所有的广告和标语一样。它们只是存在在那里，偶尔会吸引你去瞥上一眼，但大部分情况下你根本不关心。

人们使用互联网就是为了节省时间，因为这样他们可以坐在电脑前轻而易举地查询全世界的资料，而不用去跑图书馆或者专业书店。也许有 75% 的人每次打开 Web 浏览器的时候，都是从搜索引擎开始。



大部分时间里，人们是这样使用网络的：



当我们要找资料的时候，就像上面这样使用 Google 或者百度等搜索引擎，然后从（可能是海量的）搜索结果中一项一项地点击可能符合要求的超链接。

在这种情况下，用户没有心情去仔细浏览每一个网页的所有内容。他们处于一种忙碌的状态——而且他们很清楚，在绝大多数页面上只有其中一小部分内容是他们感兴趣的，至于其他的东西……“也许下次我能仔细看看”。

别担心用户们的“快速浏览”能力。日常生活中我们一直都在用这种方式快速浏览着杂志、报纸和书籍，每个人都相信自己在这方面毫无问题。

因此，人们在网页上看到的内容取决于“他们想看到的那些东西”，而不是我们呈现出来的那些东西。

我们这么设计……



用户看到的……



一般来说用户都“想”看到什么东西呢？要么就是他们手头上的任务相关的文字或短语（例如“Photoshop 教程”），要么就是当前或接下来的个人兴趣（“对了，易建联在雄鹿打得怎么样？”），还有那些在我们脑袋里根深蒂固的触发器，例如我们的名字、“免费 / 大减价”、“买一送一”还有“超靓美女”。

6.1.2　他们也不会仔细考虑

你知道 Web 浏览器中最常用的按钮是什么吗？

我们总以为用户在浏览页面的过程中，会看到所有的超链接或者按钮，然后经过考虑选择一个最合适的。然而事实上大多数时间里他们都不会这么做。

用户们总是很忙碌。他们知道“我可以慢慢寻找一个最佳的链接，但那样的话我就得花上一些时间。我为什么要花这些时间？”因此他们只是匆匆扫过页面，一旦发现一个链接，看起来似乎能跳转到他们想去的地方，OK，他们就点击了，而不会深思熟虑是不是真能达到自己的目的。

这是因为他们有 Web 浏览器的“后退”按钮作保障。

他们知道如果自己猜错了，不会产生什么严重的后果。只需要点击几次“后退”按钮就能重新来过——如果猜错 3 次以上，他们就会直接换一个网站。这也是用户们在长期的经验积累中得到的“比较明智”的方法，因为有那么多设计不佳的网站，即使他们仔细权衡了哪一个链接才是正确的，他们也很有可能得不到想要的结果……与其这样，不如多猜猜，不行的话再退回来。



这也从一定角度说明了用户为什么更喜欢载入速度快的页面——当他们点击“后退”以后，之前的页面能够很快地展现出来。如果页面载入速度比较慢，他们就需要在“后退”之后等待更长的时间。这就“迫使”他们在之前就必须更加谨慎地进行选择。用户对此总是说“不”的。

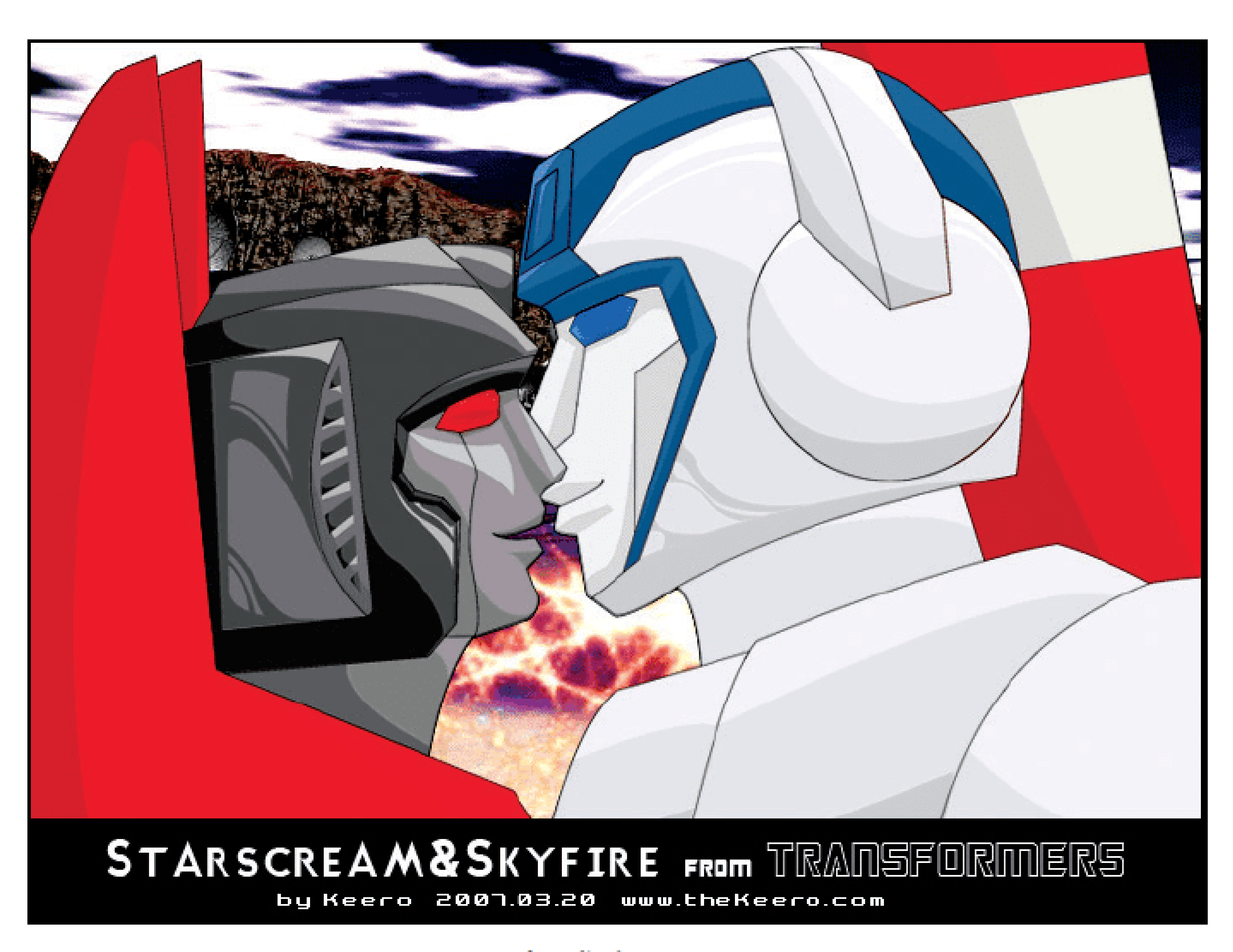
而且另一方面，猜测永远不会像仔细考虑那么“累人”。他们所要做的只是随意地点击，这样不用耗费太多的脑细胞。如果他们猜对了，速度会更快，如果没猜对——有可能我还能无意中看到某些令人意外但很不错的内容，这种机会因素带来的可能性也很让人开心。

当然我并不是说用户们在点击超链接和按钮之前从来都不进行思考或者衡量。有些人考虑得多，有些人考虑得少，但你不能因此就放弃那些考虑得少或者不考虑的用户。而且每个人在使用网络的时候都有各种客观的影响因素，例如当时的心情、具体的打算，还有时间是否紧迫。与此同时，如果一个用户信任这个网站，他可能就会多花些时间来考虑如何选择——因为他知道如果他付出一定的努力，一定会在这里得到相应的回报。

6.2　好感度计量器

什么是好感度？好感度就是当我们在电脑上玩某些角色扮演类或者养成类游戏时接触到的一个概念，用来量化男女主角之间感情的深浅。

当你做了某事，例如为对方疗伤、送个小礼物，或者拒绝其他异性的示好后，你中意的那位对你的好感度就会提高。相反，如果你是个花心大少，处处沾花惹草或者不珍惜她的感情，那么好感度就会降低，也许游戏最后的结局会让你很郁闷。



好感度……

同样的，网站和用户之间也存在着这么一种东西。我们每次进入一个网站时，好感度计量器都会有一个初始数值。如果网站让我们感觉结构清晰、很有礼貌，而且很能为我们着想，那么我们对这个网站的好感度就会上升；而如果老是遇到各种各样的问题，碰到各种各样的障碍，好感度就会逐渐降低，直到最后降为 0。

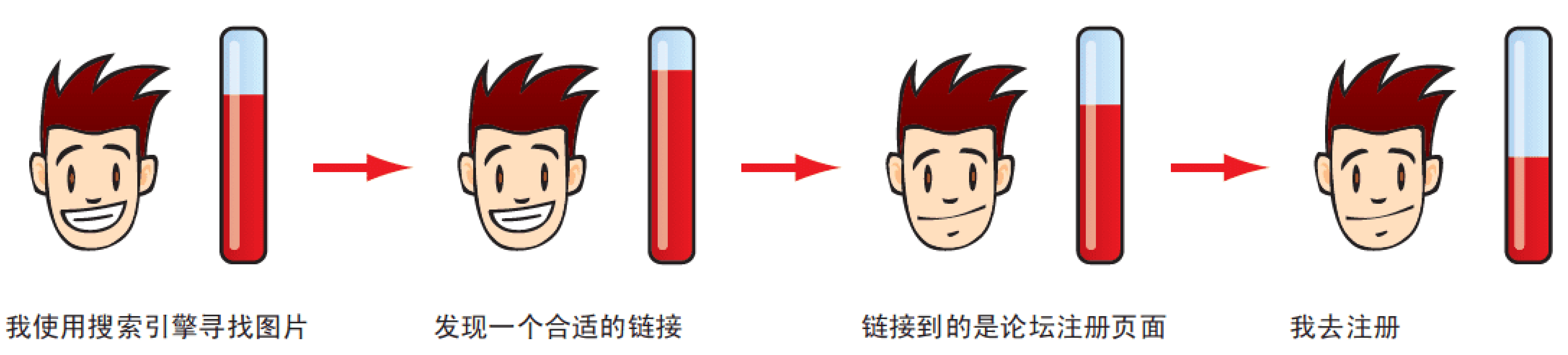
6.2.1　我经历过好几次这种情况……

在网上搜索信息或者图片的时候，会有很多搜索结果的链接指向到某些论坛的页面。可能每个人或多或少都会遇到这种情况：

中国•中文编程行业论坛-古歌计算机中文编程学校-UI设计一席谈错误信息	
您在"UI设计一席谈"的时候发生错误,共有1项,下面是错误的详细信息	
• 错误的版面参数！请确认您是从有效的连接进入。	
• 请仔细阅读论坛帮助文件，确保您有相应的操作权限。	
<< 返回上一页	
您尚未登录，您可以输入您的用户名、密码登录以便进行您的操作。	
请输入您的用户名	没有注册？
请输入您的密码	忘记密码？
Cookie 选项 请选择你的 Cookie 保存时间，下次访问可以方便输入。	不保存，关闭浏览器就失效 保存一天 保存一月 保存一年
隐身登录 您可以选择隐身登录，论坛会员将在用户列表看不到您的信息。	正常登录 隐身登录
登 录	

当你看到它的时候，是不是会想大声地叹口气？

有些时候我遇到过这种情况：

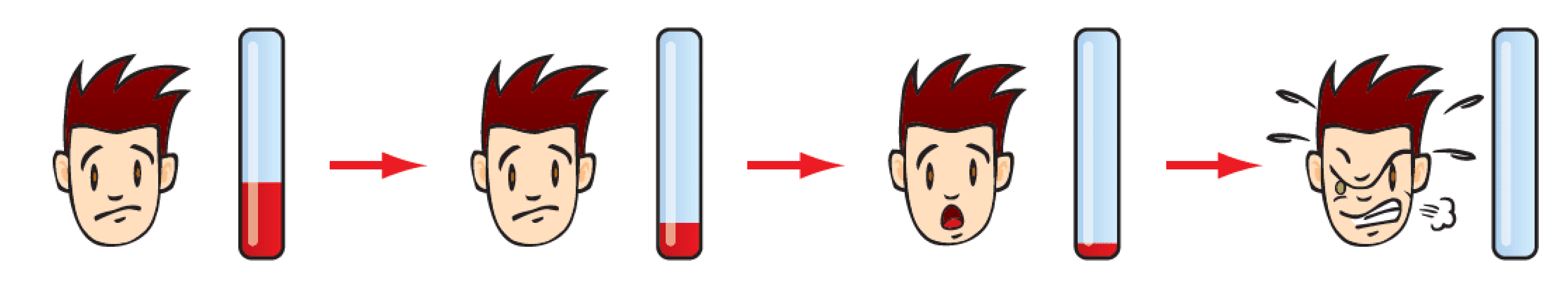


我使用搜索引擎找图片
我的好感度并不是很高，因为需要不停地点击结果页面，也不一定就能找到我想要的。

发现一个合适的链接
我就知道～哈！看来搜索引擎就是好。

链接到的是论坛注册页面
需要我注册？我从没来过这个论坛，以后可能也不会来，我不想注册。

我去注册
我确实需要这幅图片，我只能去注册。



注册相当麻烦！

用户名被占用？为什么等我点击了确定才告诉我？我总是得重复输入密码！

总算注册好了

刚才是哪个页面来着？谁的帖子？我没有注意帖子名称。论坛自动跳转到首页了。

当我来到那个帖子

“本内容跟帖回复才可浏览”？我还得想回复的内容！很多论坛禁止灌水的……

还不能马上回复！

“对不起，为防止恶意灌水，新注册用户在1小时内禁止发言”？？？！！！

没错，我知道很多论坛需要注册是为了避免服务器负担过重，或者减少“盗链”，希望通过这种方式减少一部分“无谓”的流量。但这真的很令人烦恼。你知道我有好几个朋友都以“TMDYYZC”的用户名和密码来注册这种可能只光顾一次的论坛。“TMDYYZC”——“他×的又要注册”。

好感度的容量是有限的。如果用户总是被迫因为网站的问题疲于奔命，就好像去某些高高在上的政府部门办事那样，经常碰钉子，而且得不到友善的待遇，那么他们对这个网站的好感度肯定会逐渐降低，就像现在好多人一想起来要去政府机关就会叹气一样。

而一旦好感度消耗殆尽的时候，用户就很可能会离开网站，因为这里无法再吸引他们。然而，离开网站并不是唯一可能产生的副作用，用户们可能以后也不想再来这里，或者从此也不再关注你的公司。而且经过口耳相传，可能会有更多的人对你的网站印象不佳，那么当他们光顾的时候，好感度本身可能就很低。

有些人天生比较有耐心，天性乐观，他们的好感值可能比较高，下降得也会比较慢；但也有些人脾气不好，或者喜欢猜疑，这些人的好感值就比较低，也许一个数据稍多一点的注册表单就会让他们的好感度计量器瞬间清空。这是因为各人特质不同而决定的，我们无法判断每一个用户属于什么性格的人。

另外，好感值也会根据具体情况而发生变化。比如说当用户在 5 分钟之内就必须得找到资料的时候，可能等待网页打开的几秒钟都会让他坐立不安。又或者用户刚刚在某个特别垃圾的网站得到了非常糟糕的体验，那么他在进入你的网站时好感度也不会高到哪里去（哪怕他本来的好感值比较高）。

正因为我们无法预知各种情况，所以最好的办法是：

如果好感度计量器的满值是100，
你就当每个用户在一开始都只有50以下。

如果他们确实有点心情不佳，我们可以多想点办法来让重新提高他们的好感值；如果用户好感度本来就不错，让他们感觉有人关心他们的利益也没有什么不好。

6.2.2　这些都会降低好感度！

很多事都会让用户在某个网站中得到不太好的体验，也就是降低他们的好感度。例如下面列出的一些情况。

需要声明的是，尽管我举出了很多案例，但其实这些仍然都是非常棒的网站，只是可能在某些地方有些疏漏。如果你就是这个网站的设计者，千万别生气。我希望大家能有一种互相探讨共同进步的心态。重要的是，千万别把我列为不受欢迎的访客！

让我感到郁闷

一位朋友曾经总结出上网的“四大郁闷事”：想找找不到、找到了不能点、点进去了出不来、出来了下次又找不到……

例如我进入某个博客网站，居然找遍了都没发现一个可以让我注册的地方：

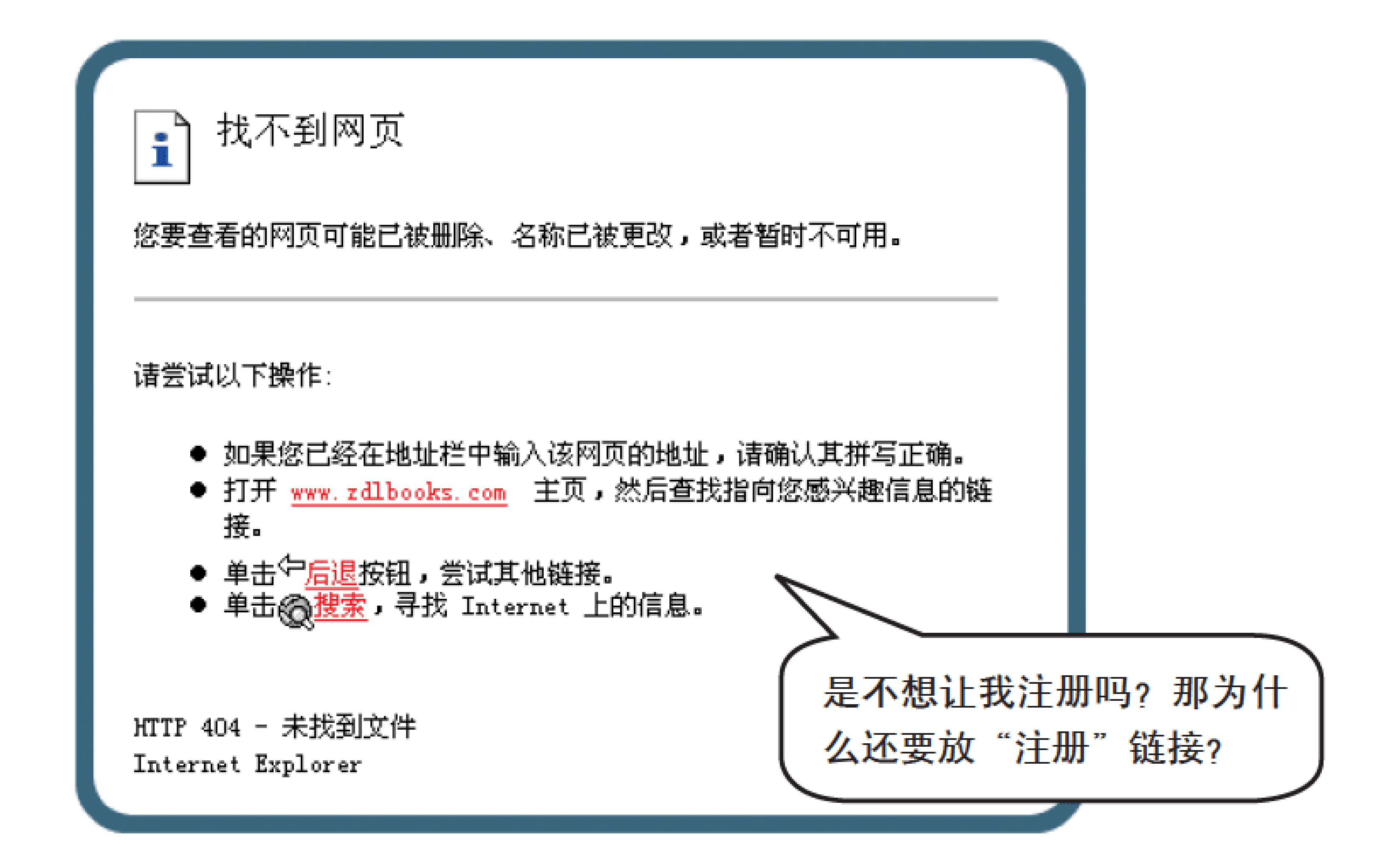


找了半天，其实是在这里，一个完全无法令人联想到“注册”的地方：

客服 你好，访客3766365 我的收藏 利海博客 积分兑换

那位朋友的结论很精辟，嗯？其实还有更多让人郁闷的事情。例如在注册和登录的时候……

最郁闷的是点击“注册”后出现这个页面：



嗯，这只是少数情况。那么更多的情况是：

创建用户并自动生成二级域名

*用户名：mess　可由中文，英文或数字组成

*姓名：弥赛亚之书

*性别：男　女

*密码：••••••　长度范围6-16位

*重输密码：••••••

*密码保护问题：我喜欢的城市

*您的答案：武汉

注册校验

*验证水印：　看不清楚？换一张

服务条款

本服务条款的生效、履行、解释及争议的解决均适用中华人民共和国法律，发生的争议提……裁委员会，其仲裁裁决是终局的。本服务条款因与中华人民共和国现行法律相抵触而导致部分无效……影响其他部分的效力。

附：
中华人民共和国电信条例
互联网信息服务管理办法
互联网电子公告服务管理规定
中华人民共和国计算机信息网络国际联网管理暂行规定
中华人民共和国计算机信息网络国际联网管理暂行规定实施办法

注册信息一经提交，即表示您同意我们的服务条款，并且已经阅读并理解我们的隐私权政策。

提　交　　重　置

为什么等我提交后才告诉我用户名已被占用？

为什么等我提交后才告诉我两次密码输入不一致？

我经常忘记勾选这个服务条款。难道就不能为我“服务”为默认接受它吗？

这是怎么回事？我已经刷新N次了。（即使有，它也老变。每次注册不成功我都要重新输入它）

上面这张图虽然是虚构的，但它表明了大多数我们经常遇到的问题。为什么不能像下面的网站这样，在我的每次操作之后系统都能即时检测和判断？这样我在填写注册信息的时候就不会心惊胆颤了！



总是误导我

一个朋友喜欢听相声，有一次他向我抱怨某个相声网操作很麻烦，总是不能多个连续播放。我去那个网站后发现确实有个“连播”按钮，但却是置灰的状态，鼠标悬停在上面时也没有任何变化。但其实它却是可以点击的（还包括最右边那个不起眼的“全选播放”按钮）。想想看有多少用户会被这个“灰色的”按钮所误导（最左边是唯一貌似可以点击的按钮，它的功能是“全选”）。



看看下面的登录栏，不知道你怎么样，反正我总是觉得那个“注册”按钮有某种磁性，“勾引”着我的鼠标不自觉地往那儿移动。我总是要提醒自己：“我已经注册了，只用点‘登录’就可以了！”



总是让我猜

好多网站总有些地方让人摸不着头脑。

看看下面的导航栏，你知道这些栏目都是些什么意思吗？



再看看这个：



“一些文字”，“多一些文字”？“再来一些文字”？……可能设计者觉得这些用词比较有意思，但我不明白它们每个的功能到底是什么。为什么有的是一般文本框，有的却又能像 Word 那样编辑？它们之间有什么区别？……咦，怎么还有两个保存的按钮？

总是要我找

我想下载“媒体中国 - 设计联盟”，你能找到下载的链接在哪里么？



我想下载这个矢量图。点击图片变成了下一幅图片，“右键另存为”又只能保存作为示意图的 JPG 文件。到底该点哪里才能下载到我想要的矢量图？



找了半天，其实是在这里：



总是说我错

也许我的智商不是最高，但也不至于这么不客气吧，全是“失败”、“红叉”，还兜头给我那么多感叹号！！！看来我真的是个差学生？



居然还奚落我？



就算我好不容易答对了，也不夸奖一下 ：

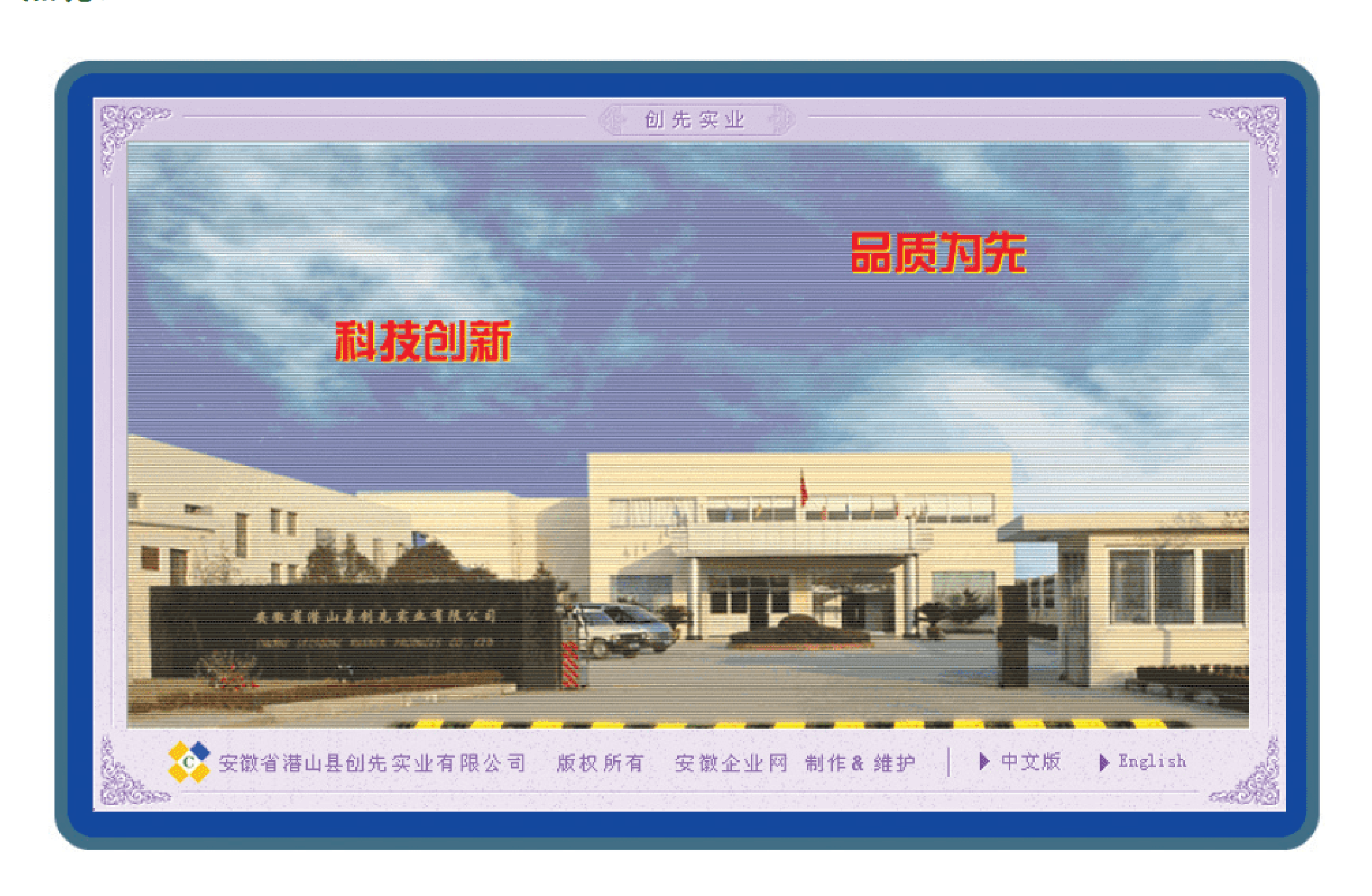
用户名： UCD
可是中、英文、数字，下划线及短横线中的組合。
检测用户名
用户名不合法！！

用户名： UCD_Z
可是中、英文、数字，下划线及短横线中的組合。
检测用户名
用户名可用

耽误我的时间

我真的厌倦了长长的 Flash 网站片头，还有里面那些超大尺寸的图片……设计者知道等待它们下载完成需要多长时间吗？我知道他想展示他们企业的风姿，但很明显这个公司并不在乎客户忙不忙，他们有多少时间。

在下面这个网站中，漫长（而且不太好看）的片头动画没有任何的“跳过”按钮。直到等它结束了很长一段时间之后我才发现原来整个动画本身可以点击，之后即能进入网站主页。似乎设计人员考虑到了人们并不太愿意老是看动画，但是为什么不能把“跳过”的功能标识得更明显一点呢？



甚至还让我抓狂

不知道你有没有过这种经验：在论坛或者 blog 里面写了一大篇文章，点击“提交”却告诉你字数超过了限制，之前写的就此消失。之后你努力回忆全部重写，并且严格确认字数，再次提交，居然因为服务器忙而导致页面错误，或者看到打开的页面是“警告：网页已经过期”……

你辛苦数小时的成果就这样付诸东流了。现在你是再重新写一次呢还是就此放弃呢？

以上都是一些降低用户好感度的例子。我们每天可能都会浏览无数的网页，这些情况可能并不少见，而且还有更多的例子我没有枚举。

当然有些时候，特别是出于一些商业上的原因，我们也许会“被迫”降低一些可用性，做一些对用户不太友好的事。例如在页面上增加广告，或者使用自动弹出窗口。这些确实会让人们反感，但如果能够证明你可以通过这些增加 10% 的收益，而且你觉得这样得罪用户很值得，那么你完全可以增加广告或自动弹出窗口。不过事先要确定，你是有意这么做的，而且有充分的理由，而不是不负责任地瞎来。

6.3　因此，为扫描而设计

我们知道用户使用互联网都有自己的习惯。如果不考虑他们的这些习惯，他们对网站的好感度就会降低。

他们总是随意地扫视页面，就好像开着车从一个个页面旁边疾驰而过，对大部分内容看都不看一眼。

也许你会觉得，我花了那么多时间设计网页的方方面面，每个按钮，每个表格，每个字段。既然浏览者对它们都视而不见，这么做还有意义吗？这样看来我还不如把时间用来做点公益劳动，起码那样还有人看得到我的付出。

但事实并不是这样。如果你认真考虑了可用性，用户们仍然能够感觉到你对细节的关注。毕竟“好用的”网站才可能保持用户对网站的好感，从而愿意继续光临。既然他们不太注意你想要他们注意的地方，那么你就提醒他们注意好了；既然他们都是扫描页面，那么我们就为了扫描而设计吧！

要做到“为扫描而设计”，你就需要保证用户能够在有限的时间之内看到你希望他们看的地方，而且帮助他们尽量轻松地浏览网页。这里有几个方面需要你注意。

6.3.1　更清楚的布局和视觉层次

网页内所有的内容都应当清晰可辨。而且你应该明确表明内容与内容之间的逻辑关系——哪些

内容相互间有关联，哪些内容又是其他内容的组成部分。这可以说是网页设计最根本的依据。要做到视觉层次清楚，就需要做到以下几点：

越重要的内容越醒目

要想使某些内容更突出，就得让它有以下特点：

- 更大的字号
- 更粗的字体
- 更特别的颜色
- 更接近页面的顶部
- 旁边有更多的空白
- 或者以上所有特点

你需要让它们成为浏览者视觉的焦点。

我最重要！

我不那么重要。

我不重要。

应当避免把重要内容放在屏幕的右侧或下端。这些地方往往被用户所忽视。而且如果浏览器的屏幕较小，用户必须使用滚动条才能看到它们。

相关的内容看上去也得相关

例如相关的内容可以分为一组，采用同样的字体和样式，同时放在同一个区域里面。

网络游戏
电脑游戏
电视游戏
电子竞技
手机游戏
动漫秀
攻略秘籍

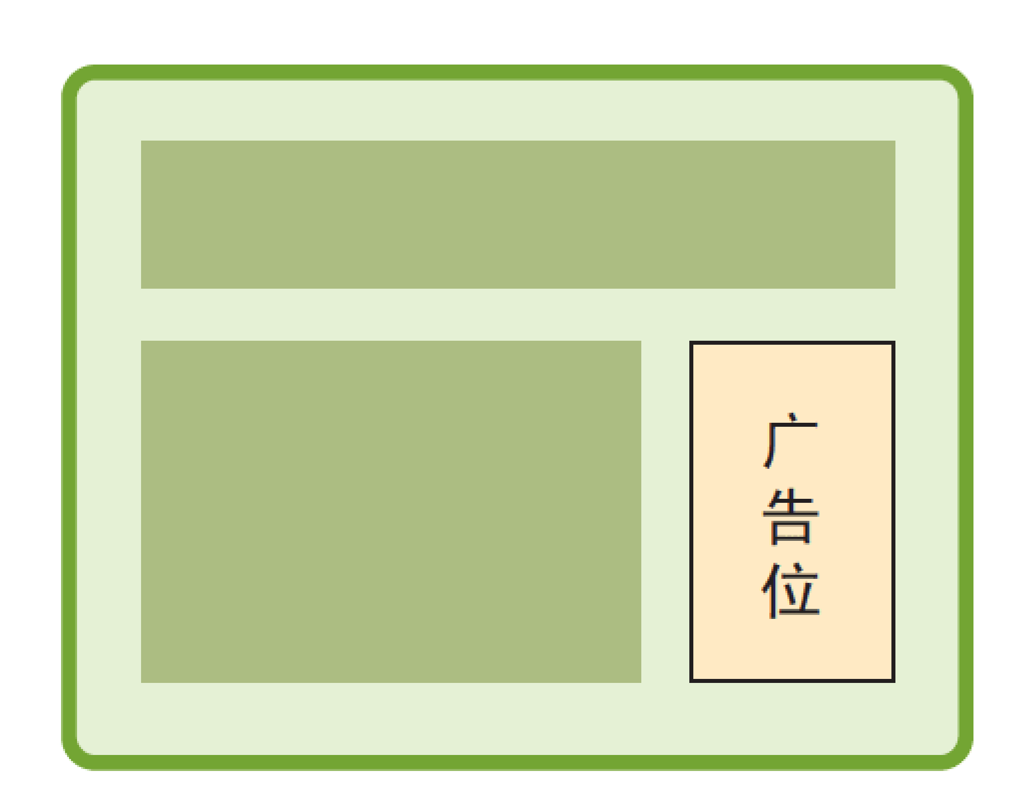
包含的内容要显示从属关系

你可以采用字体大小、颜色、缩进等方式来表达这种从属关系。同时嵌套也是一个很有表现力的方式。



别显示太多无关的信息

我们要明确每个网页的宗旨，把网页的大部分以及最有效的显示区域用来显示与网页宗旨直接相关的信息。版面不是主要用来放广告的。广告可以放，但别鸠占鹊巢。



其实我们每天看的报纸就是一个非常好的例子。报纸总是有很清晰的视觉层次，否则阅读者根本无法从密密麻麻的文字中找到阅读的线索。报纸的排版都使用了突出、分组和嵌套的原则，这些原则帮助了我们。其实在阅读的时候我们根本没有意识到正在“分解”这些视觉层次。而一旦它们不够清楚（例如所有内容看起来都差不多重要），你就会“被迫”减缓扫描的速度，以自己的方式去寻找线索——这就是为什么有些报纸读起来感觉很累的原因。

新疆石油报

责任感造就影响力

依靠自身力量进军煤层气勘探领域

克拉玛依油田首次在和什托洛盖地区实施大型压裂

最重要的文字和较次要的文字。

2006年10月31日 星期二 农历丙戌年九月初十 总第7269期 （本期16版）

这些内容都是并列放置在这里，它们都是一些常用信息。

"时代杯"头条新

这些内容都属于"敬告读者"，因为它们是嵌套放置的。

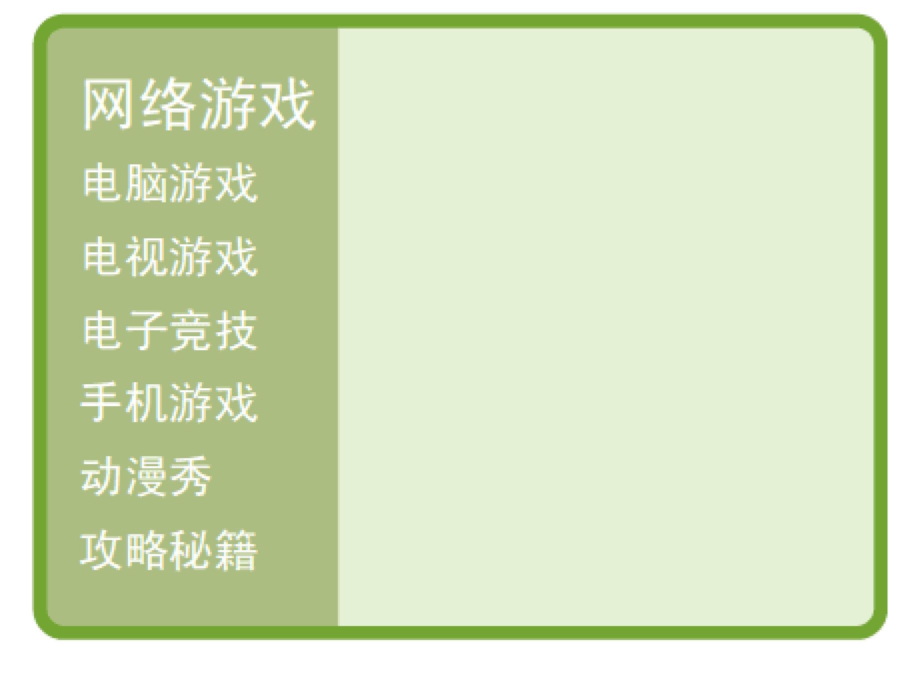
试油公司将首次用上自编HSE培训教材

敬告读者

这三列文字很明显都属于一篇报道，因为它们位于同一个标题之下。

采油二厂油藏目标化，"管"

在某些导航机制中，有可能你会突出某个栏目名称以表明现在用户所处的位置。这是一个很好的方法。但是同时也要注意，不能因为过分突出某个栏目而拉大它和其他栏目之间的距离，导致出现"嵌套"的局面。例如下面这种情况：



其他类别都属于网络游戏吗？

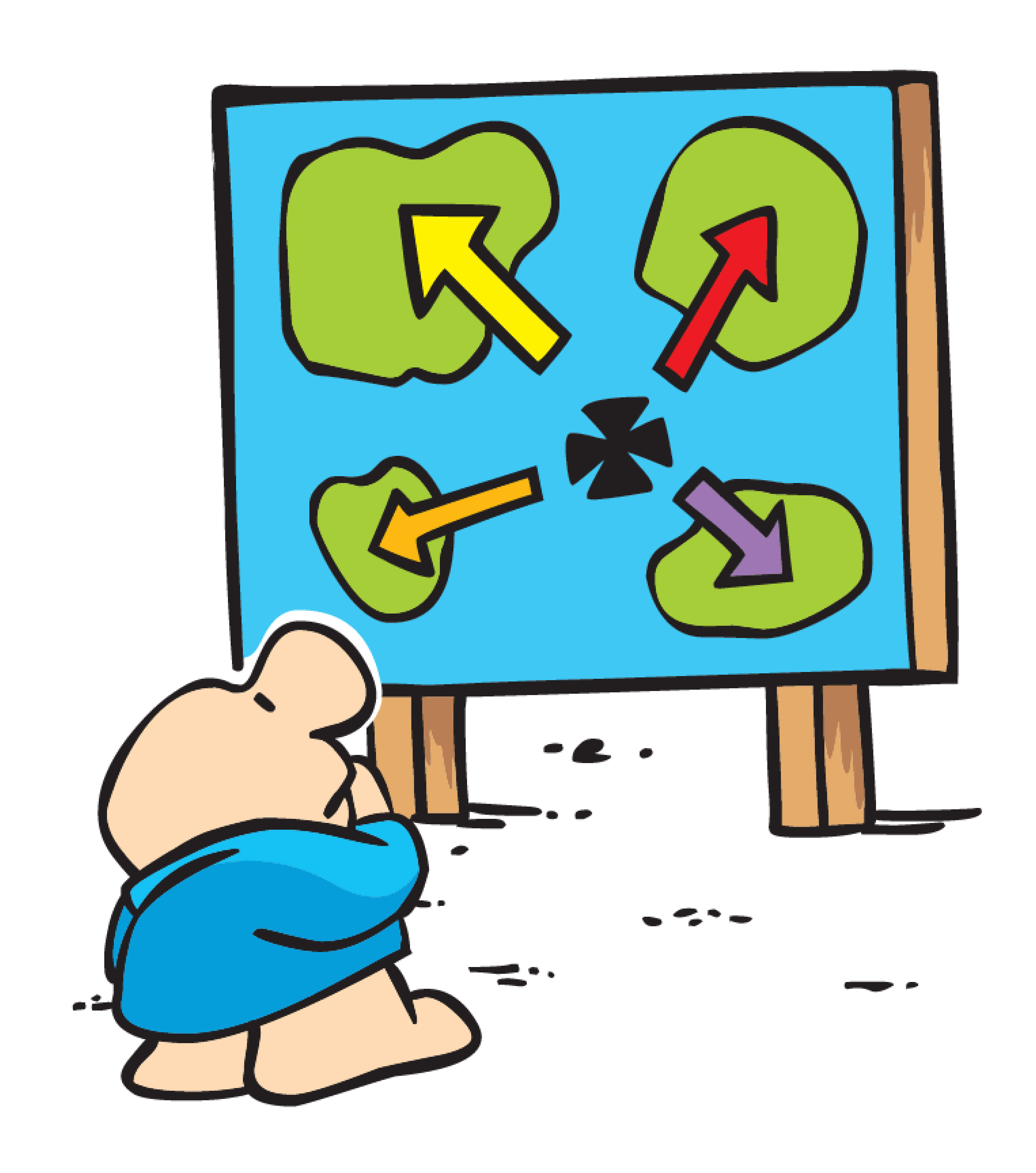
网络游戏
电脑游戏
▶ 电视游戏
电子竞技
手机游戏
动漫秀
攻略秘籍

这个就好多了。

6.3.2　清晰地划分页面区域

因为用户总是很快速地浏览和扫描，你就需要把页面划分成明确定义的区域。网页的各个部分可以分为导航栏、主要内容区、图片区、广告区，各个区域之间可以使用一些视觉处理方法予以区分，相互间不能混杂。

用户只要短时间的扫视之后，应该就能指着页面的不同区域说："我能通过这里前往其他地方"、"这是今日的头条"、"这里我能看到一些感兴趣的产品"。这样便于他们很快地决定应该关注页面的哪些区域，同时可以放心地跳过哪些区域。



另外，你也可以多使用清单和列表。人们在浏览整齐排列的清单或列表内容时，速度往往都会明显快于阅读连续的文字。例如下面的网站：



6.3.3 适应用户的习惯

用户们在长期使用互联网的过程中已经积累了很多经验。很多网页对他们来说都是不言而喻的，因为他们已经很熟悉网站的普遍使用方法。所以，在设计网站时一定要保持通用性和实用性，并且记住下面这句话：

用户大部分时间都是使用其他网站的。

就像阅读报纸一样，了解版面的布局和格式的习惯用法能让我们更快、更容易地扫描页面。我们也能很容易地阅读其他城市（甚至国家）的报纸，因为它们都采用差不多的习惯用法。



即使一个字也看不懂，我们也能大致知道有些什么内容，以及如何使用

用户不可能只使用我们设计的网站，更多的时间他们都花费在其他网站上。他们也不一定愿意花费时间和精力来接受我们的“特殊”设计。用户想得到的信息和服务并不一定只能从一个网站里得到，别忘了你有相当多的竞争对手。如果用户到达一个从没光顾过的网站，不能在几秒钟时间内觉得网站的内容合乎他们访问的目的，或者说不能清楚地了解网站的使用方法，他们很可能马上掉头就走，尝试其他的网站。

所以，有时候更新颖的网站设计，往往也同时伴随着较高的危险性——它有可能迷惑用户。

但是设计师们通常都不愿意“拘泥”于原有的习惯用法。他们觉得自己之所以是设计师，就是要做一些崭新的、与众不同的设计，而不是使用那些原有的东西。没有人会因为“他保持了用户的习惯用法”而得到赞扬或者奖励。

而且我们并不是否定创新。有句话叫做“Having a healthy disregard for the impossible.（理性地藐视不可能的事)”。如果你认为某种新奇的设计与习惯用法一样易学易用，或者认为它能够带来非常大的价值，值得用户付出一点努力来学习，那么不妨大胆地辅助实用。这就像创可贴能取代绷带一样。但前提是：你确实认识到它的价值大于习惯用法的价值。如果不能确定，多做点用户测试。如果 50% 以上的人都认为它比较糟糕——那就还是回到习惯用法上去吧。

6.3.4　提供目光的“落脚点”

在真实的世界中，要记住行走的路线往往都是利用一些突出的、与众不同的景物或建筑标志。例如“到了招商银行后往左拐进一条小巷子”。与此相似，你也可以在网页的不同位置提供容易辨认的表达形式，这样同样能够帮助用户扫描和记住网页的内容。

例如，在以文字为主的网页中，在适当位置加入图标或分隔线，以及具有醒目字体和颜色的标题等等，这些都可以成为用户区分不同内容的有效标记。它们能够为用户把视觉焦点从一个地方移动到另一个地方提供线索。

看看下面的文本，你的视线是否感觉不太容易集中?

“红帽不仅是Linux，它将以中国开源领袖的角色出现在中国市场。”8月27日，红帽大中华区总裁胡伯林对记者说。在今年7月，胡伯林作为红帽大中华区新一任掌舵者接替了原红帽大中华区总裁陈实的职务。红帽方面表示，胡伯林的就任是大中华区新篇章的开始。
政府是最重要的客户
红帽是全球最大也是最知名的开源软件企业，这些年在中国的业务也发展得很快，但是政府采购市场几乎是红帽的一个“死角”。红帽在中国的主要竞争对手是中科红旗，中科红旗由于有一定政府背景，在政府采购中屡屡得手。而红帽则成为惟一没有入围中央国家机关协议供货的在京开源软件企业。
对此胡伯林坦言，在政府采购等领域的表现“几乎为零”，但是他上任以后情况将要发生变化，未来红帽大中华区的策略将分成行业、政府、服务三大部分。其中政府将是最大最重要的客户。
重塑品牌形象
分析人士表示，红帽在政府采购市场没有什么成就主要是两个问题造成的。首先是销售模式“水土不服”，红帽的销售模式基本上是“订阅模式”，而政府采购则通常采用招标的形式，“订阅模式”无法适应政府采购制度；其次，红帽的知名度太低，对政府采购人或采购代理机构来说，“红帽是做什么的”依然是一个需要普及的常识问题。
对此胡伯林表示，红帽要与政府部门进行一些沟通，把红帽的想法告诉他们，并听取一些政府用户的意见，然后再决定未来的销售方式。红帽希望把产品的整个生产及架构解释得更清楚一些，让政府采购重新认知红帽。
至于红帽原有的订阅采购模式，胡伯林表示红帽也不会轻易放弃，因为这种模式确实能够给政府带来好处，目前多数软件公司一般采用许可销售方式，新产品发布之后，往往会通过终止对老版本的销售来达成强制升级，从而带来更多销售收入。而使用订阅模式，用户可以免费试用产品，觉得对自己有价值再付钱。尽管现在政府可能不习惯这种采购方式，但是在了解他的好处后未必不能接受。
胡伯林告诉记者，红帽在全球并不缺乏经营政府行业的经验。在印度，红帽50%的生意是政府带来的。
根据红帽的经验，要想赢得政府采购，首先要说服政府使用开源软件，现在微软等专有软件产品已经发展得非常成熟，而政府采购人更愿意使用成熟的产品来建设他们的信息系统，因此必须让政府用户意识到开源的重要性、开源的价值。
此外，胡伯林也坦言，红帽自身也缺乏政府采购行业的知识，需要进行“恶补”。
培养专业人才

哪怕我们让每一段文字之间多一个回车，稍微加大点行间距，可能就要好得多。

“红帽不仅是Linux，它将以中国开源领袖的角色出现在中国市场。”8月27日，红帽大中华区总裁胡伯林对记者说。在今年7月，胡伯林作为红帽大中华区新一任掌舵者接替了原红帽大中华区总裁陈实的职务。红帽方面表示，胡伯林的就任是大中华区新篇章的开始。

政府是最重要的客户

红帽是全球最大也是最知名的开源软件企业，这些年在中国的业务也发展得很快，但是政府采购市场几乎是红帽的一个“死角”。红帽在中国的主要竞争对手是中科红旗，中科红旗由于有一定政府背景，在政府采购中屡屡得手。而红帽则成为惟一没有入围中央国家机关协议供货的在京开源软件企业。

对此胡伯林坦言，在政府采购等领域的表现“几乎为零”，但是他上任以后情况将要发生变化，未来红帽大中华区的策略将分成行业、政府、服务三大部分。其中政府将是最大最重要的客户。

重塑品牌形象

分析人士表示，红帽在政府采购市场没有什么成就主要是两个问题造成的。首先是销售模式“水土不服”，红帽的销售模式基本上是“订阅模式”，而政府采购则通常采用招标的形式，“订阅模式”无法适应政府采购制度；其次，红帽的知名度太低，对政府采购人或采购代理机构来说，“红帽是做什么的”依然是一个需要普及的常识问题。

对此胡伯林表示，红帽要与政府部门进行一些沟通，把红帽的想法告诉他们，并听取一些政府用户的意见，然后再决定未来的销售方式。红帽希望把产品的整个生产及架构解释得更清楚一些，让政府采购重新认知红帽。

另一方面，使用正确的对齐和排版方式也很重要。比如说，减少用户眼光水平扫视的范围，他们阅读或扫描的速度便会大大加快。你可以把较宽的文字段落以较小的宽度显示。此外，大多数情况下书写的习惯是居左对齐，但在用户填写表单时，有时候把项目名称居右对齐反而可能使用户更容易地与相应的文本框对应。

例如下面右边的表单，在填写时就比左边的要相对更容易辨识一些。



6.4　同时，减少用户的思考

糟糕的设计是把简单的事情复杂化。

平庸的设计是把复杂的事情复杂化。

而优秀的设计则能够把复杂的事情简单化。

人无法停止思考。我们已经说过，用户思考是很有必要的。每当他们来到一个网站，他们都会思考以下几个问题："这是什么地方？"、"这里有我需要的内容吗？"、"我还能去哪里？"。

但有些思考就是不必要的了。

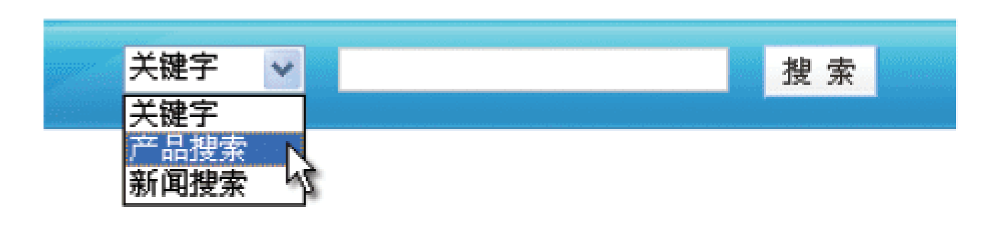
在浏览网页的时候，网页的每项内容都有可能迫使我们停下来进行不必要的思考。我们的脑中会不断地出现各种问号，把注意力从要完成的任务上引到别处。英语有句俗话叫"monkeys on my back"，这些问号就像不安分的猴子，也许一两只还能够承受，但如果它们越来越多，压在我们背后就会足以令人抓狂。

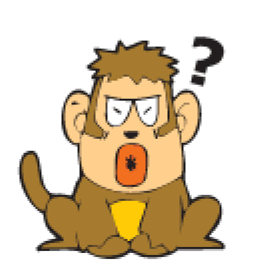
要想去掉这些问号，做到"减少用户的思考"，有几个方面需要注意。

6.4.1 减少不确定因素

在用户光临你的网站之前，可能从没有接触过你的网页。当他们在网页中浏览或选择时，可能会有很多不确定性。我能点这个按钮么？我选择"香港"地区时打开的网页是中文还是英文？这是网站内的搜索还是整个互联网的搜索？很多时候他们不敢去尝试，害怕由于错误的选择而出现意外的页面，从而耽误他们的时间。

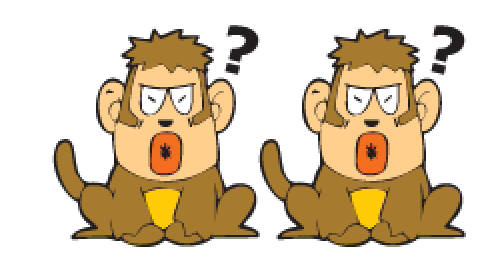
这些"不确定因素"产生的问号会出现在页面的各种地方。在之前"降低好感度"那一节我已经举了很多例子。当然还有更多的例子：



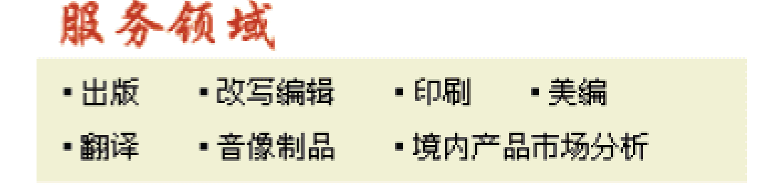


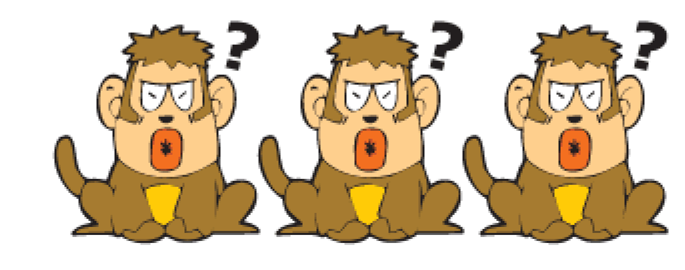
如果我不选"关键字"的话，是否就不能输入关键字了？



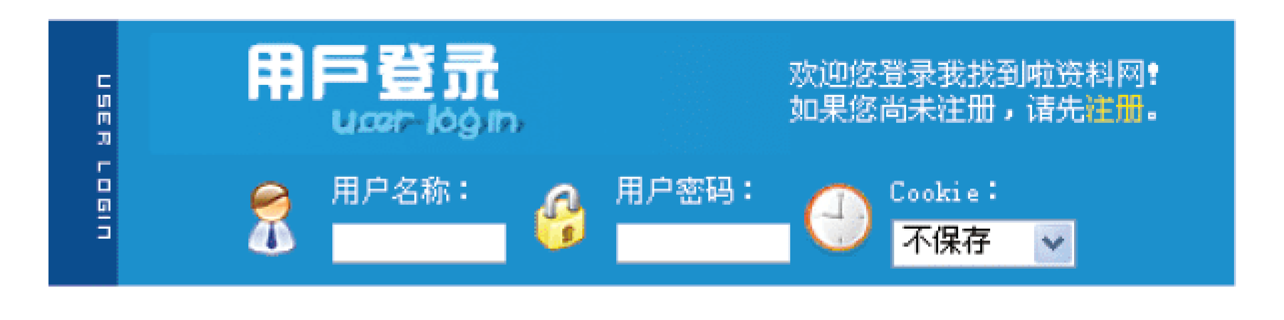


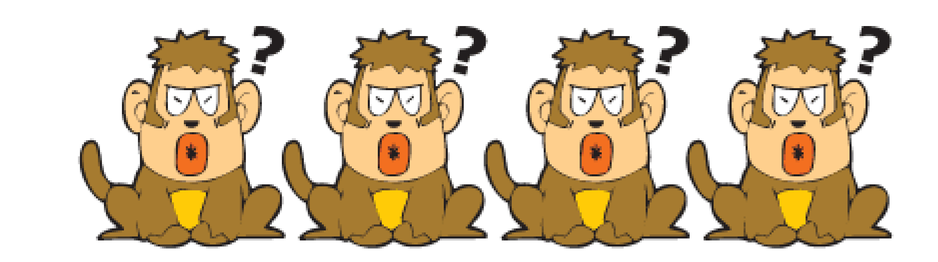
我现在到底在"首页"还是"我的当当"？





这些到底能不能点击？（结果是不能点击）





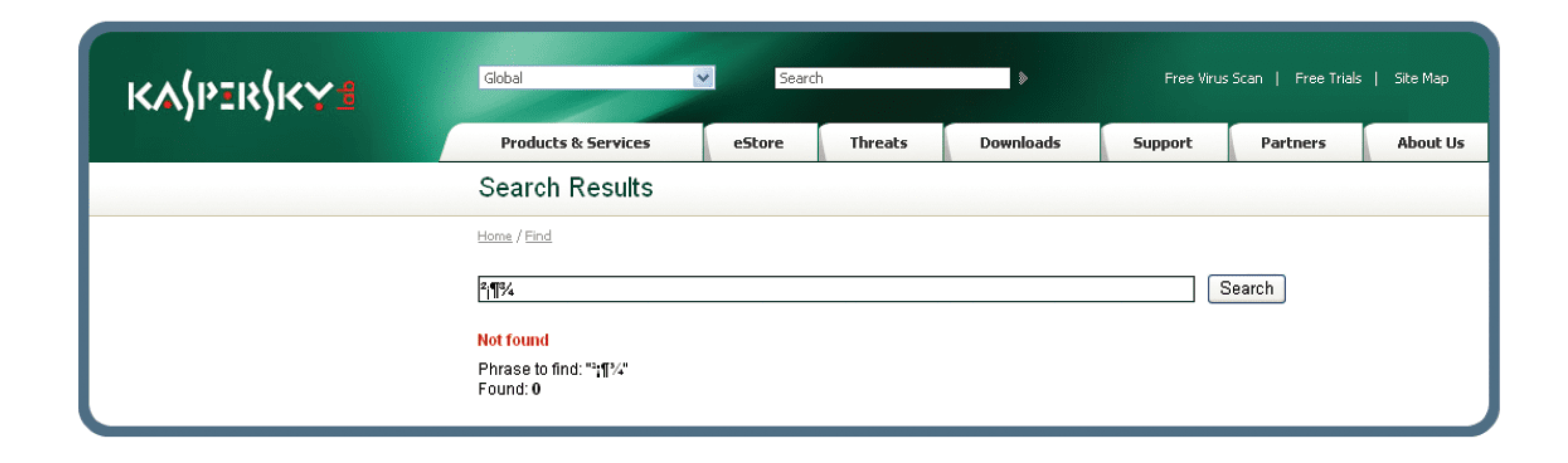


Cookie是个什么玩意？另外，我点什么进去啊？

卡巴斯基实验室的网站在这方面做得就很不错。按钮和超链接清晰可辨，各个国家和地区的下拉列表选项中也标明了页面所用的语言。不管我是个人还是企业用户，也不管我是想试用、购买还是在线扫描，都能明确地知道通过什么操作能达到自己的目的。



可惜的是，当我放心愉快地在“搜索”区域输入“病毒”之后，得到的结果让人很沮丧：



为什么会这样？可能的解释是，卡巴斯基实验室的搜索程序只认识英语（或俄语），它把中文当成乱码了。但很奇怪的是为什么会连到它的国际网站上去（你看我现在已经开始不必要的思考了）。不管怎么说，我再也不敢在卡巴斯基网站上用中文搜索了。我同时开始隐隐担心的是，它的中文网站里会不会还有别的链接让我莫名其妙地跑到国际网站上去呢？

其他还有很多地方都需要减少用户的“不确定”问号。例如你可以对链接进行必要的解释。要知道有些链接用户并不一定想点开。

有的网站会提供 DOC 或者 PDF 格式的文件供用户下载。如果没有相应的提示，有时候用户会点击 PDF 文件的链接。通常这个时候 Web 浏览器会出现一个新的窗口，下载并显示一般来说比较巨大的 PDF 文件。这可能会占用很大内存。许多人都不习惯这种方式，他们会说“早知道是这样，我还不如直接下载到硬盘再打开呢！”

有些网站注意了这个细节，在下载链接前面提供了文件格式的图标。这样用户可以马上了解链接文件的格式，从而做到有选择的下载。



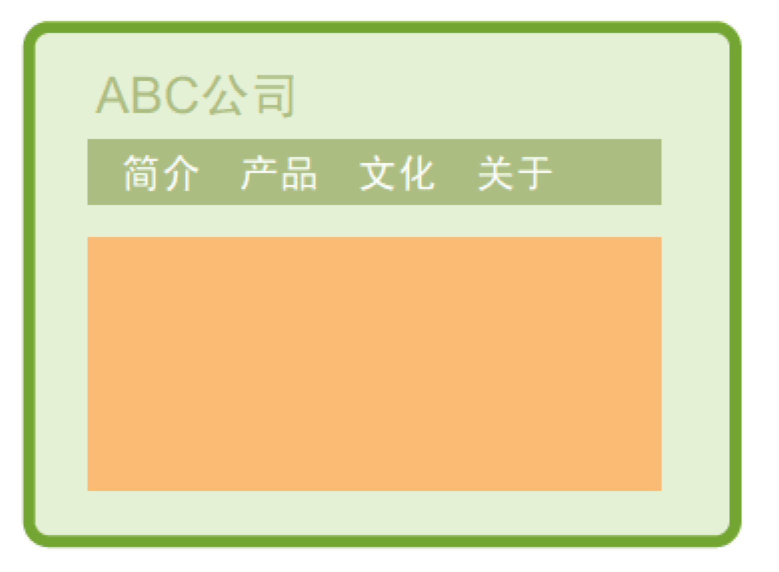
6.4.2 保证网页的一致性

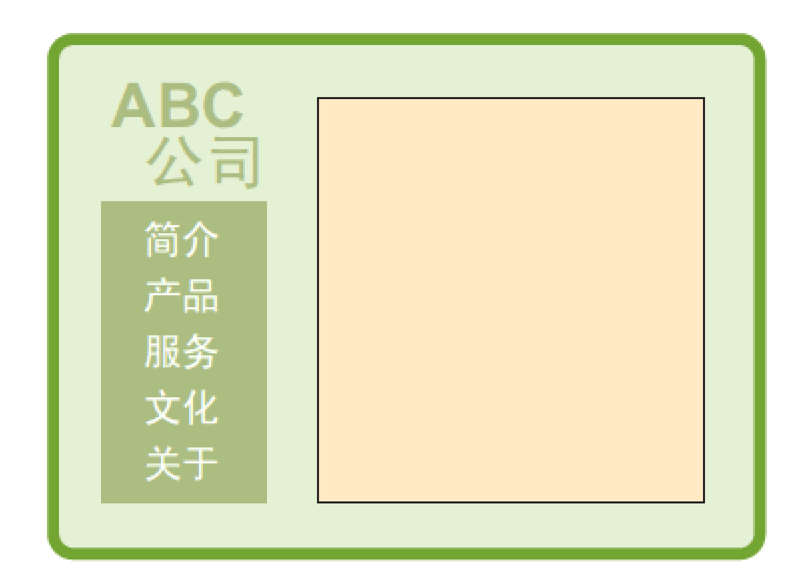
一般情况下，一个网站里面所有页面的设计风格需要保持高度的一致性。具有高度一致性的网站能给用户清晰感和整体感，便于他们“扫描”，也能够用这种“确定性”减少他们可能产生的问号。同时，浏览网站的时候用户能够感受自己身处一个经过精心的全面设计的“空间”之内，一切都井然有序。这也能提升他们的情绪值和好感度。

我在第三章曾经描述过一些具体的设计指南，里面提到过“一致性”的问题。现在看看对于网页来说它应该如何解释。

一致的网站标志和导航栏

它们可以说是最体现网站特征的部分。这就像 VIS（视觉识别系统）一样，应该对 logo 的颜色、大小、位置等方面都有严格的说明。如果一个网站里面每个网页的 logo 都大小不同，或者位置不统一，或者导航栏在某些页面突然多出两个子选项，给浏览者的感觉一定很不好。



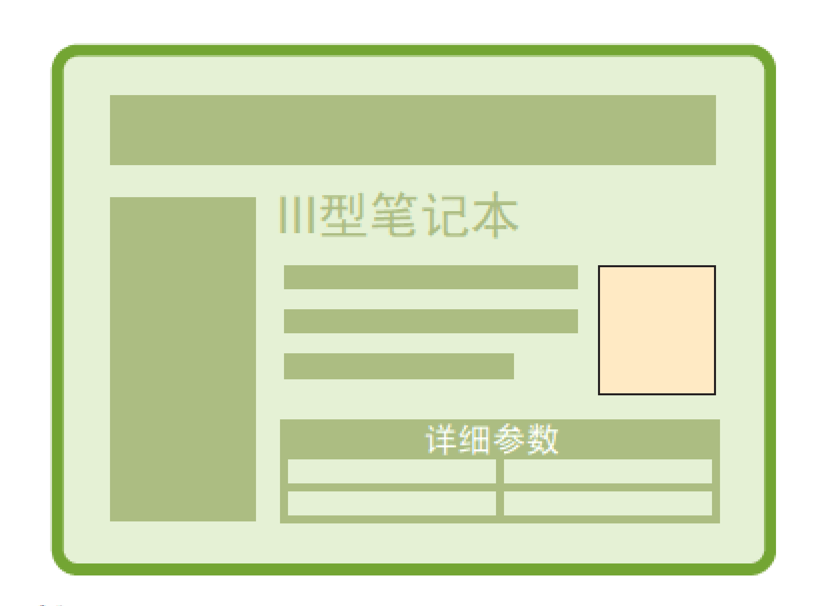


这是一个公司的网站吗？

一致的页面布局

这可以说是一种“包装”，我们使用图片或者图表来表达一些信息。如果某些页面的等级是一致的，那么它们的“包装”风格也应是一致的，用户应该使用同样的浏览方式来阅读这些网页。





这是一个公司的网站吗？

一致平衡的信息结构

网站每个栏目的信息量应当合理分配。除了一些特定的内容（例如“网站地图”和“联系方式”），所有分支的信息量应当相对平衡。如果用户在 A 栏目中看到有 20 多个下级栏目，但 B 栏目中却只有 3 个，他们肯定会有点头昏脑胀——你迫使他们去思考“为什么会这样”。

一致的重复性元素

如果使用了文本框，那么网站中所有文本框的形式都应当一致。如果某个图标的尺寸是 40×40 像素，那么相类似的图标大小都应该差不多。

一致和谐的字体和色彩

字体和色彩也应遵循同样的标准。你可以为不同的版块设置不同的色系，但是如果你这个版块用的是橘红，那么它所有的页面都应该是橘红。版块与版块之间的差异也应该明显。

以上是一些考虑一致性的例子。如果想保证一致性，一个很有效的方法就是在设计网站时制定正式的“设计风格标准”。这种标准规定了网站中所有网页都必须遵守的设计规则。“设计风格标准”

并没有固定的格式，但应当非常具体，包括页面所使用的图标、文字、表格等各种元素的颜色、大小、格式等内容。

一致，但不一样

需要注意的是，一致性并不是“相似性”。如果太相似，很可能用户无法区分彼此之间的不同。例如下面的网页，图标之间的颜色、形式过于相似，尽管图标能代表各选项的功能和概念，但用户一眼看上去无法觉察出差异，图标没有体现出明确辅助标识的作用。



6.4.3 明显标识可以点击的地方

人们在网络上做得最多的一件事情就是找到下一个地方点击。如果一个地方是可以点击的，那么你就应该明确地告诉他们。

也许你会说："哪里是可以点击的——我把鼠标移上去，如果箭头变成了一只手，那就表示可以点击。这并不需要费多大功夫。"但是如果我们迫使用户去思考本来通过下意识就能知道的东西，这纯粹就是在浪费他们对新网站本就很有限的耐心和好感。

如果是一个按钮，那么我们不需要去担心。用户们知道按钮都是为了点击而设置的。但如果是文本超链接——我们知道，现在几乎 80% 的网站都不再使用缺省的超链接形式了。也就是说，不仅超链接的文字不是蓝色，而且好多超链接根本就没有下划线。有些超链接文本在鼠标移上去的时候出现下划线，有些则只是变个颜色。

但这些也不会造成太大的影响：我们使用了这么多年的互联网，已经具备了一定的经验，能够识别出什么是超链接。

那是否就是说这个问题可以不用考虑了？当然不是。那到底要注意些什么呢？

关于文本超链接，我们需要把握一个原则：

让用户清楚地区分超链接文本和其他文本形式，并潜意识知道可以点击。

具体来说，我们可以分三种情况来讨论：

有时候下划线可有可无

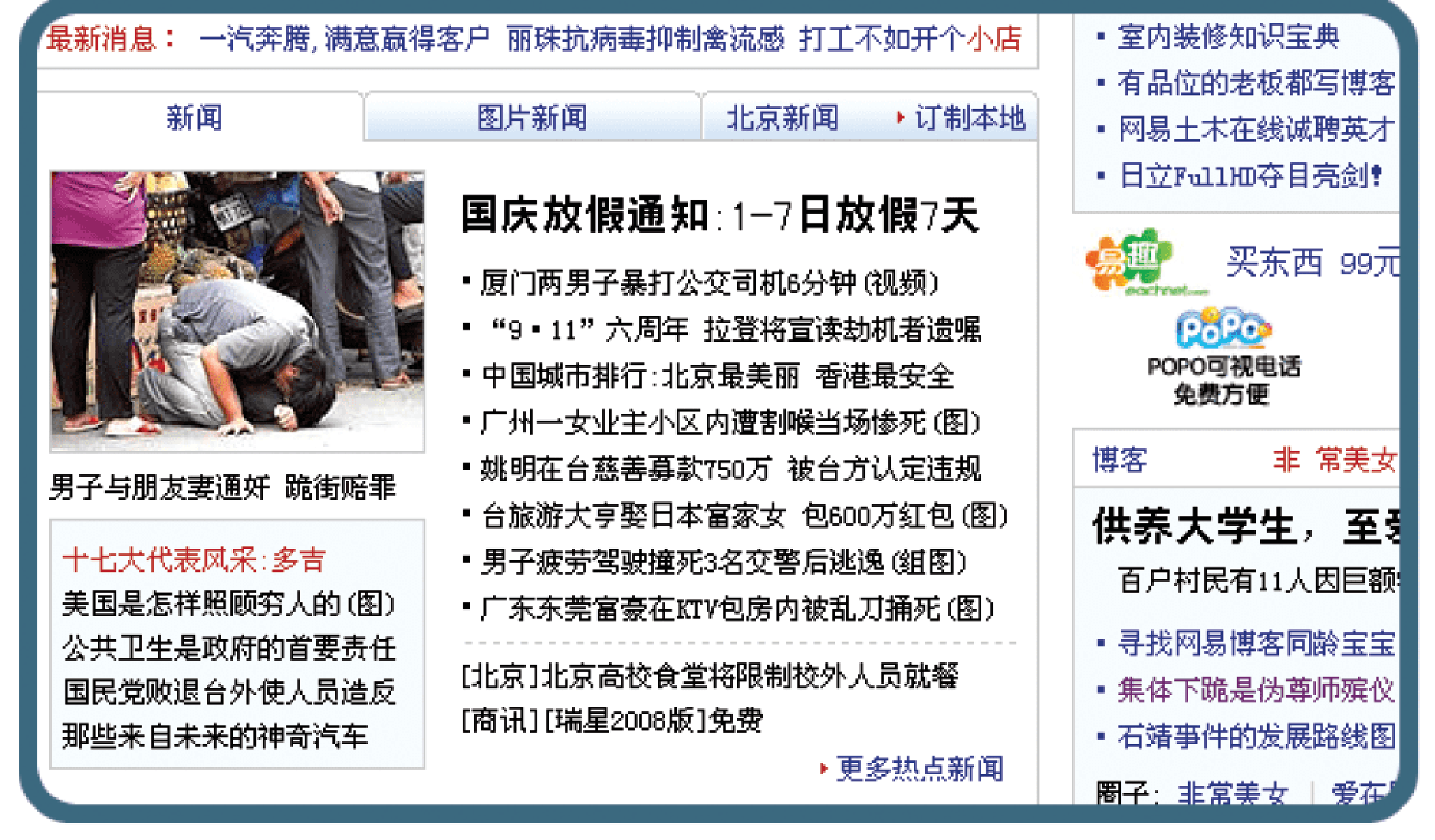
如果是网站的导航区域，我会潜意识去点击里面的任何东西——这就是经验告诉我的。所以不管里面的文字链接是否带有下划线，对用户来说都没有什么影响（只要不是有的带有的不带就行）。



还有一种是整个页面全是超链接的情况。很多门户网站，例如新浪、搜狐，首页里面的内容几乎全部都是可以点击的。这种时候这些超链接可以有下划线，也可以没有。例如网易的首页就去掉了超链接的下划线。



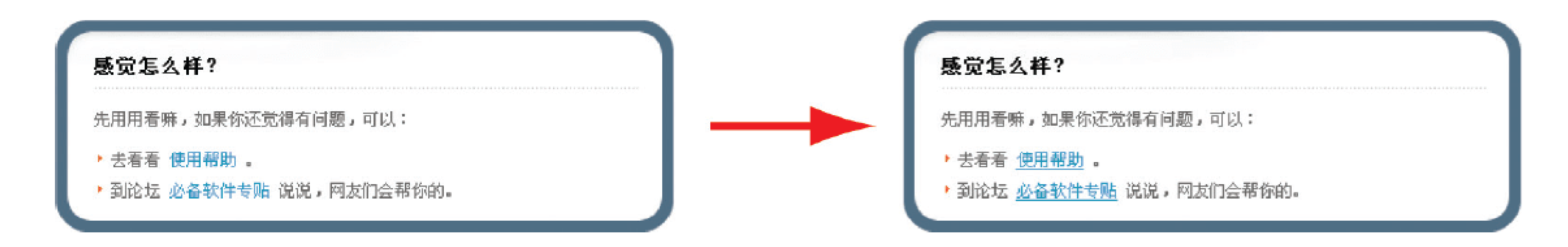
搜狐的首页全部都带下划线



网易的首页则没有下划线

有时候最好有下划线

一般来说，在文字链接和普通文本混排的情况下，下划线更容易让我们去点击。在这种时候，普通文字的表示方法最好明显区分于超链接。



另外，如果是独立的文字链接，也需要加上下划线，不然很容易被人忽略以为不能点击。



超链接和按钮

有时候你也可以把文字链接做成按钮的形式，以此来吸引用户的注意。也许很多人不太了解文字链接和按钮的区别，或者说不太清楚什么时候该设置为文字链接，什么时候可以做成按钮。

一般来说，“文字链接”在点击后只是链接到一个新的页面，而“按钮”则更富功能性，例如表单提交和下载，它们涉及到数据的转换。用户点击了链接后不会出现什么严重的后果，随时能够通过浏览器的“后退”按钮回到前一个页面；但点击按钮就是确定提交了数据，这个时候浏览器的“后退”是没有用的。

但是由于现在互联网使用习惯的演变，对于网页来说，超链接和按钮之间的差别已经不大了。对于用户来说，它们都是可以点击的，点击后会得到某个结果。用户不会去区分什么是按钮什么不是按钮。对他们来说，重要的是“页面上的内容变了”，以及“我要的东西出来没有”。所以从这个角度上来说，不管是超链接还是按钮，你只需要注意其表现形式的统一性即可——如果需要突出这个链接，把它做成按钮的样子也未尝不可。

6.4.4 清晰、简单的网页内容

大多数用户并不会把网站用作阅读工具。由于屏幕大小和下载要求等因素的限制，用户一次只能看到一个网页的一个屏幕范围内的显示内容。他们使用网站的典型方式是通过浏览一些比较简单的信息而决定下一步要做什么。

当然有一种例外，即在线文章或新闻网站。用户浏览这些页面的时候有着明确的目的，他们就是冲着这些文字来的，而且做好了心理准备——我要沉下心来，好好读读到底是怎么回事。这种特例不是我们讨论的内容。

用户在阅读其他媒体时，例如书籍和报纸，往往会比较容易和迅速地专注在某一特定内容上而开始连续的阅读。但是他们在使用网站的时候，则会把很多时间和注意力集中在选择和决定上。如果我们的网站上文字累赘、艰涩难懂或者夸夸其谈，这都会妨碍用户的使用。

关于网页内容的编写，有一些需要注意的地方：

使用用户的语言，而不是技术语言

我们要考虑到网站浏览者们的水平、品味和知识结构。并不是所有人都是专业人士，所以要避免使用普通用户不能理解或不习惯的语言，例如专业名词、方言和英文字母缩写。

现在很流行一种被称为“RSS”的内容共享方式。但很多人根本不了解什么叫做“RSS”。如果看到某个网站上有个这样的图标，人们可能会不知所以，尽管它非常简单实用。



这是个什么玩意？

其实“RSS”（Really Simple Syndication，真正简单的整合）是一种描述和同步网站内容的格式，是目前使用最广泛的 XML 应用。通过 RSS 每个人都能自由调用（或订阅）提供 RSS 服务的网站内的一些信息，或者提供自己发布的信息让其他站点共享。

也许在按钮下面加上几个字就能减少很多用户的疑惑：



哦，我可以订阅他写的日志

与之相似的还有下面这个按钮：

+ RSS 订阅本站各栏目

使用通俗的语言，而不是故作高深

平易的语言使读者能轻松而快速地理解内容，因此不要使用不必要的复杂或抽象语言。文绉绉的“官方用语”总让人要“转个弯”才能想清楚。

例如，“撰写”可以考虑改成“编写”，甚至更简单地改成“写”；“雷同”可以改成“相同”；“最终”可以改成“最后”。这样我们能够拉近与用户之间的距离，而不是摆出一副冷冰冰的“客套”面孔。

减去不必要的词句，让页面更简短

不必要的文字简直就是在浪费用户的时间。在很多网站上面，大部分文字都是在占地方而已，没有人打算读它们。去掉没人会看的文字可以让有用的内容更突出，也可以让页面更简短。用户在页面上一眼能看到更多的内容，他们滚动屏幕的次数也会少一些。

例如，“以往的经验”直接改成“经验”就可以，“互相协作”也应该变成“协作”。多出来的那些字词就是我们所说的“废话”了。你应当看着页面上的每个句子每个词问一遍“这里是否可以去掉？是否可以更简短？”如果是，毫不留情地减掉它们。

此外，多使用短句会使用户阅读起来不那么费时费力。因为复杂的长句往往让人迷惑。

不要夸夸其谈，或者让人摸不着头脑

网站的用语应当谨慎得体。很多网站太过于注重宣传和推广，极度美化自己的公司或产品，而且使用的词汇抽象空洞。这些我们在日常生活中的广告里面经常能体会到。这些夸夸其谈的内容根本不会给用户带来任何有价值的信息，相反，它们会妨碍用户的使用，甚至带来反感。

另外有些网站提供一些莫名其妙的内容。用户看到这些后只会摸不着头脑，甚至会“鄙视”网站的所有者。

某个网站的“公司文化”栏目里面，通篇在介绍该公司所在城市的历史古迹、文人墨客，还提供了很多诗词歌赋供浏览者欣赏。但是关于企业自身文化方面的宣传倒少得可怜。这就有点本末倒置了。



最下面的那段文字应该去掉。人们看不出京剧鼻祖程长庚和公司的关系——如果实在想要，把它移到第二段之前，效果也会比现在好一些。

6.5　告诉用户他们在哪里

互联网是一个特殊的空间。人们游弋在其中，抵达一个个网站，在其中的一些页面上落脚，获取信息，然后再跳到其他网站或页面上去。

没错，是“跳到”，而不是一步一步走过去的。我们通过“点击链接”向前移动，没法看清楚路上有些什么东西。我们也没办法记住它的物理位置（根本也没有什么物理位置），如果希望再次访问某个内容，我们只能重新顺着以往的链接找过去。

在这个奇怪的空间里面，你感觉不到网站的大小（除非它特别小），根本不知道这个网站还有多少页面没有打开过。同时你也没有方向，所有的页面只有上一级下一级的层级关系，没有哪个页面在这个页面的左边或者右边。

所以当我们置身网络时，根本就是迷路的。在这里没有 GPS 导航。如果你不去为用户考虑这

些方面，他们很可能就会在网站里左寻右觅却找不到需要的内容，或者根本无法摸清这个网站是怎么组织的。那么他们也不太可能在我们的网站里待很长时间——因为他们需要自己去理清脉络，找寻线索，这种思考太费劲了。

因此，为了提高网站的可用性，让用户能够以更高的效率完成任务或者达到目标，你必须让他们时刻清楚自己身在何方。

6.5.1　导航栏存在的意义

看看现在的互联网。不管是商业网站还是个人网站，也不管网站的风格是简洁还是繁复，是卡通还是写实——不管这些网站是什么样子，它们都有导航栏。



为什么每个网站都有导航栏?

这就好比我们逛商场找不到电梯和洗手间的标志，高速公路上看不到出口和加油站的预告牌，或者翻书没有目录、章节标题和页码一样。我们会感到不方便，无法定位。

设想一下没有导航栏的情况吧。

- **我们不知道当前的位置**

这一点在前面已经说过了。网络是一个奇怪的空间，没有导航就好比没有向导，我们没有稳

定的感觉，只能自己摸索。你觉得你是更喜欢迷路还是脚踏实地了然于胸？

- **我们找不到想要的东西**

没有导航栏，我们无法知道网站的所有内容。仅有的线索只有页面里的那些超链接——而从数百个超链接里找到真正想要的可能非常费劲。另一种途径是搜索，但是如果我们不太清楚要找的关键词，甚至网站根本就没有提供搜索工具，那也是毫无用处。

- **我们不知道如何使用网站**

如果有导航栏，我们能够知道从哪里开始，能进行什么选择。但如果没有，我们只能自己在字里行间去寻找——估计没几个人会有时间和心情玩这种游戏。

- **我们对网站很容易失去信心**

如果对网站无法建立起一个比较清楚的概念，我们对它的印象肯定也好不到哪里去。要让我去猜吗？网站的建设者知不知道自己在做什么？如果连他们都不知道自己在做什么，我还会再来这里吗？

导航栏的存在还有一点非常重要的理由，那就是我们几乎想象不出来没有导航栏的网站会是个什么样子。这说明导航栏的概念已经深深植入我们的脑海里，它已经形成了一种习惯，而习惯成自然。那么，如果没有更好的创意，我们最好还是服从用户的习惯吧。

6.5.2 导航栏所需要的元素

一般来说导航栏都应该包括 5 个元素：



网站 logo 标识

我们凭借网站的 logo 标识来识别目前所在的是哪个网站。它就好像是徽标或者商标，有其独立的识别性和显著性。我们应该一眼就能分辨出一个 logo 标识和其他 logo 标识的区别。由于网站的特殊性（便于用户扫描的浏览方式），网站 logo 标识不应过于复杂或者花哨。一个很难看清有些什么的网站 logo 标识不算一个好标识。

网站中的每一个页面最好都显示网站的标识。这样我们才能够在任何时候都能明确自己所在

的位置，因为大多数情况下我们都是通过超链接进行页面间的跳转——“嗖”的一下我们就瞬间移动到某个页面上去了，如果没有清晰的标识我们可能会摸不清方向。



一般来说网站标识都在页面的左上角。这是因为这个标识图案或者文字就代表着整个网站，它的结构层次最高，涵盖着整个页面所有其他的元素。

网站的栏目

顾名思义，栏目就是网站结构化后的各个子版块。不同的版块决定了彼此不同种类的内容，因此也应该涵盖它所包含的元素。



对于那些比较大的网站来说，可能仅有大的栏目还不足以对内容进行分类。为了便于浏览者理清网站结构，我们还需要对栏目进行二次分类。这也就是常说的“二级导航”。



“回到首页”

这是导航栏中非常重要的一个元素。也许你不知道该怎样去往网站的其他地方或者使用其他的功能，这时候只要点击“回到首页”就能够瞬间回到最开始的地方。不管你是否迷路，你都能随时重新开始。



另外，网站的 logo 标识也应能起到相同的作用。让它成为“回到主页”的按钮毫不费事，只需在图片的属性上加一个链接地址即可。这样一来我们又有了一个新的保证——不费力，但很有效果。

附加功能

这里可以说是一个实用工具栏，它同样指向一些重要的页面，但这些页面并不属于网站的主要内容。它们一般是一些针对网站本身所需要的功能，例如帮助、网站地图，或者是一些关于网站发布者的信息，例如“联系我们”等等。



这些功能或者信息都非常重要，所以我们必须要放在比较首要的位置。但另一方面，这些实用工具只是一些附加的功能，所以它们也不应该过于显眼，至少不能像栏目那么显眼。

此外，由于导航栏那里空间有限，如果放太多附加功能可能会有点喧宾夺主——起码也会让用户眼花缭乱。有时候你也可以把它们中的一部分放在页面的最底端。例如：



搜索工具

很多人喜欢搜索工具。有些人一到某个网站就开始寻找搜索框，就好像他们一到商场就找服务员一样（这类人大多是男性）。对于那些大型网站尤其如此，用户没有（或者不愿意花费）很多精力和时间去自己寻找内容，特别是当他们觉得自己要找的内容比较“冷门”的时候。



搜索工具要注意的是，清晰地告诉浏览者“搜索工具在这里”。不要使用自以为是的用词，例如把搜索按钮写成“找你所找”或者“我帮你查查”之类的词汇。这个地方不要抖聪明，因为使用搜索工具的用户要么就非常直接，要么就是迫不得已，这些人不会因为这里的小幽默而开心——相反，这些词迷惑了他们，要不就是耽误了他们的时间。

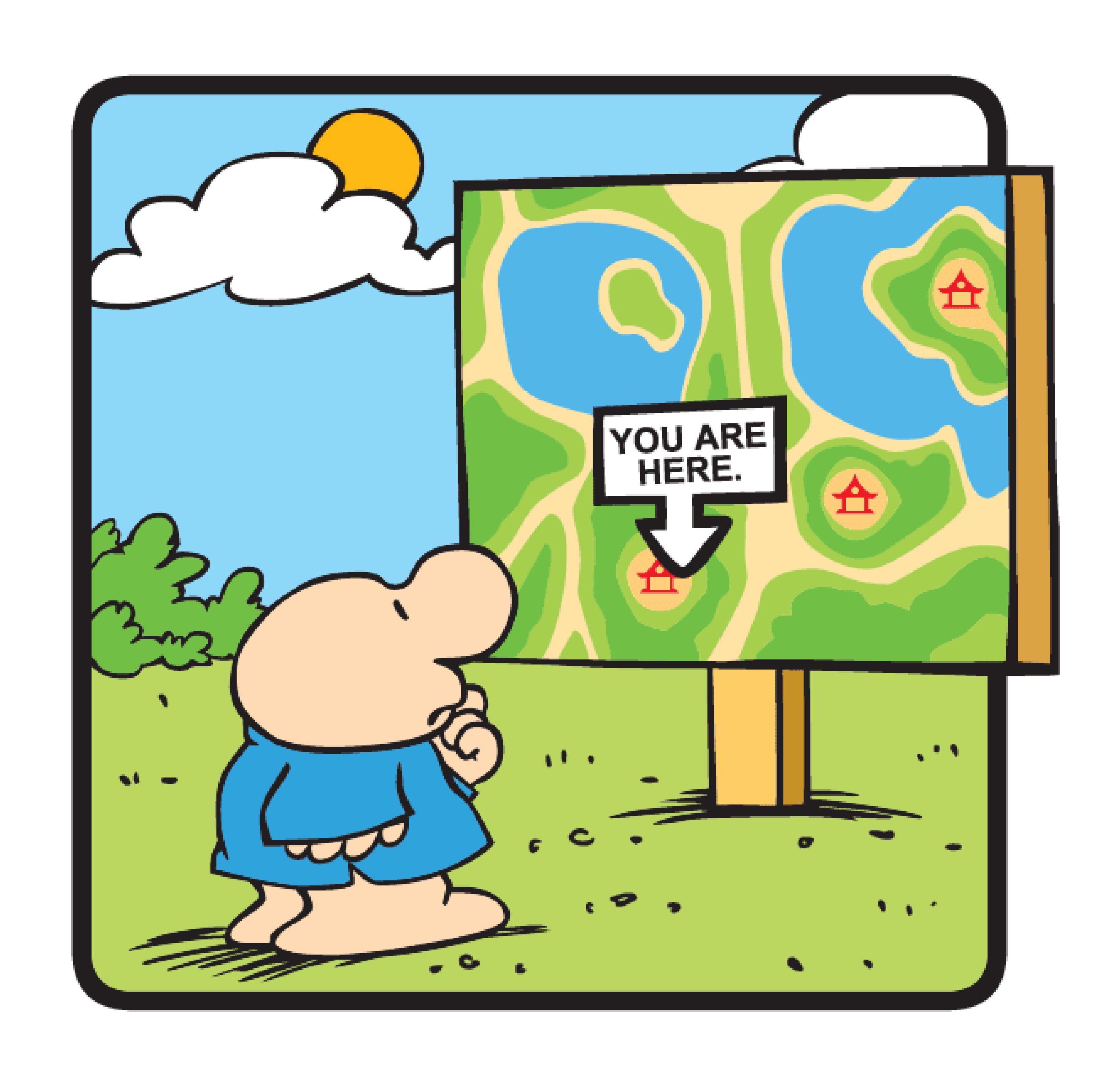
6.5.3　导航指示、页面标题和“面包屑”

导航栏可以说是一个工具，浏览者使用这个工具能够非常清晰地掌握网站的大体结构，使用一些必备的工具。这样能够非常有效地提升用户的稳定感，也便于他们找到想要找的东西。

但除此之外还有其他一些需要注意的地方，通过它们你能更周到地为用户指明方向。这些地方包括导航指示、页面标题和“面包屑”。

导航指示

所谓“导航指示”，就是在导航栏中标记当前页面的位置。它的作用就好像在公园或者住宅区里面的地图指示牌，上面一般会有一个“你在这里”的指示箭头，我们能够很轻易地知道现在何处，去往某处又该怎么走。



通过这种方式，我们能在导航栏中一眼就看到自己所处的位置，找到自己此刻是在哪个栏目或者版块内，同时要前往其他栏目应该如何点击。



导航指示的标记方式有很多，也非常灵活，我们唯一需要注意的是让它更为突出和明显。但如果没有导航指示或者标识不清，用户就容易摸不着头脑。

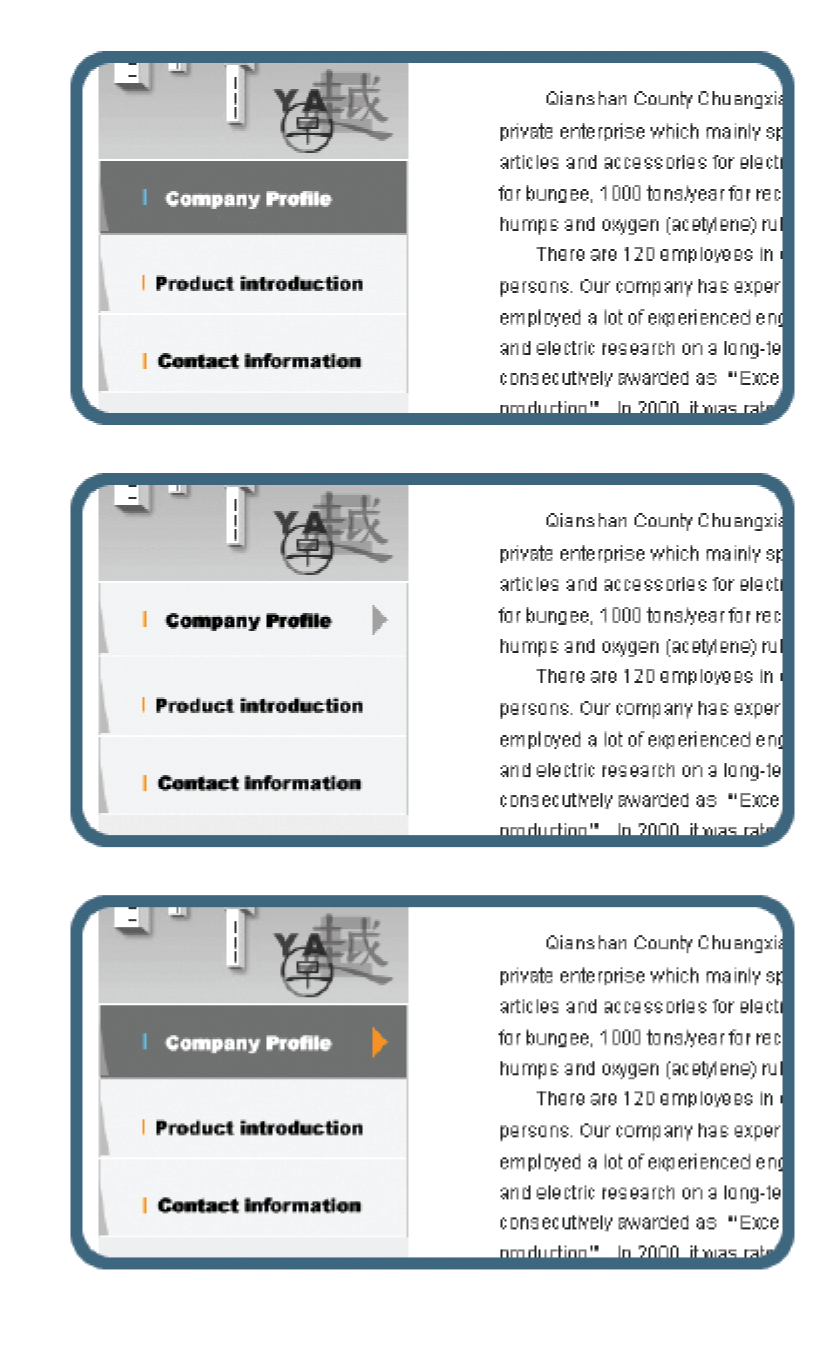
例如下面这个网页：



我现在在哪个页面？

其实仔细看是能发现导航指示的：左侧导航栏中“Company Profile”的字体比下面两个栏目稍微大了一些。但这个标记太不明显，几乎没有人能够注意到这里的变化。

其实我们有很多种方法可以让这里的导航指示更为明显。例如把导航栏中该栏目按钮反白显示，或者加上箭头，或者既反白又加上箭头。



页面标题

光有导航栏的指示是不够的，因为大多数网站中一个栏目里面很可能不止一个页面。我们应该让浏览者随时知道目前阅读的是哪方面的信息，或者正在浏览哪一篇文章。这就好像杂志中每篇文章都有一个标题，我们每翻过一页后，通过标题才知道“哦，现在我看的是 × × 文章”。



正因为是一个页面的标题，它就应该比较引人注目。这可以通过字体的大小、粗细、颜色以及周边的留白来体现。英文网站可以把标题文字设置得很大，在 Windows 的 Web 浏览器中也能有很好的表现力，但是中文就难说了。不过你可以通过为标题添加一些附属元素（例如背景）来表现出它与一般文字之间的不同。

另外，页面标题应该涵盖整个页面的内容，最好放在正文的上方。



“面包屑”

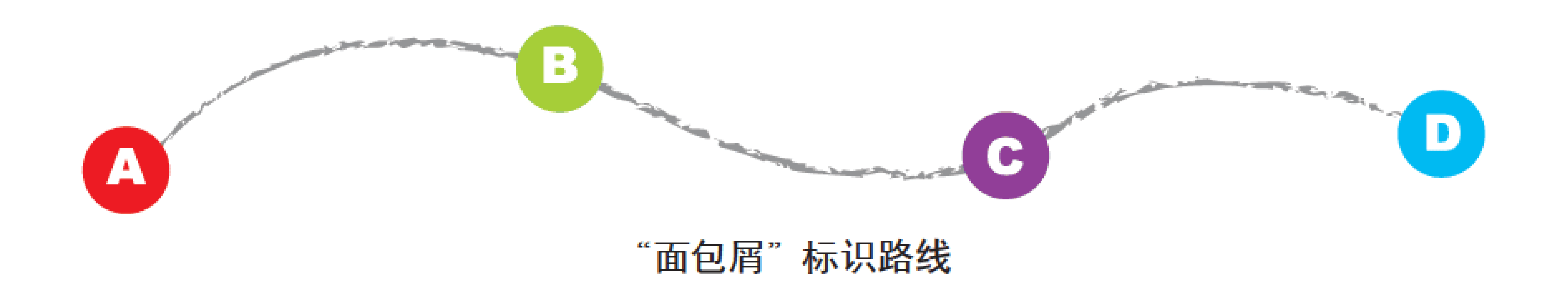
我不是在开玩笑，页面中大多数时候确实需要它。它的正式名称也就叫做“面包屑”，英文名为“Breadcrumb”。这是一种层级菜单。

之所以称其为“面包屑”，是因为在格林童话《亨赛尔与格莱特》中，两兄弟为了避免迷路，在森林里沿途撒面包屑以便标识回家的路线。

其实我更偏向于把这种层级菜单称为“忒修斯之线”。这是因为格林童话里的面包屑很快就被小鸟吃掉了，最终兄弟俩没能回到家，反而落入了一个巫婆的陷阱。但是在希腊神话里面，勇士忒修斯要去迷宫杀死牛头怪弥诺陶洛斯，他把一个线团的一端拴在迷宫的入口处，然后沿途放开线团,最后顺利完成任务并且循着线找到了出口。这个典故似乎比“面包屑”更为合适。但可能“忒修斯”这个名字在我国没有多少人了解。既然大家都叫它“面包屑”，那么最好还是遵从用户的习惯比较好。



可以看出，“面包屑”也从某个角度告诉了我们当前的位置。与导航指示所不同的是，它只告诉了我们从主页到当前位置的路径，而导航指示则告诉我们这个网站的层级结构。可以说导航指示类似一张地图，而“面包屑”则是一个简单的路线图。我们需要知道路线，但是地图能够告诉我们更多。



“面包屑”标识路线

“面包屑”可以说是一种经济实用的手段，因为它不会占用太多空间，而且非常方便。用户能够随时做他们最常做的两件事：回退到上一层，或者去主页。对于那些大型网站“面包屑”尤其有用。

早期的“面包屑”中分隔层级的符号使用的是冒号“:”或者斜杠“/”，但后来逐渐变成了大于号“>”。这种演变体现了一种视觉上的暗示，表示出每一层级的大小关系，而且有一种逐渐向前移动的韵律感。

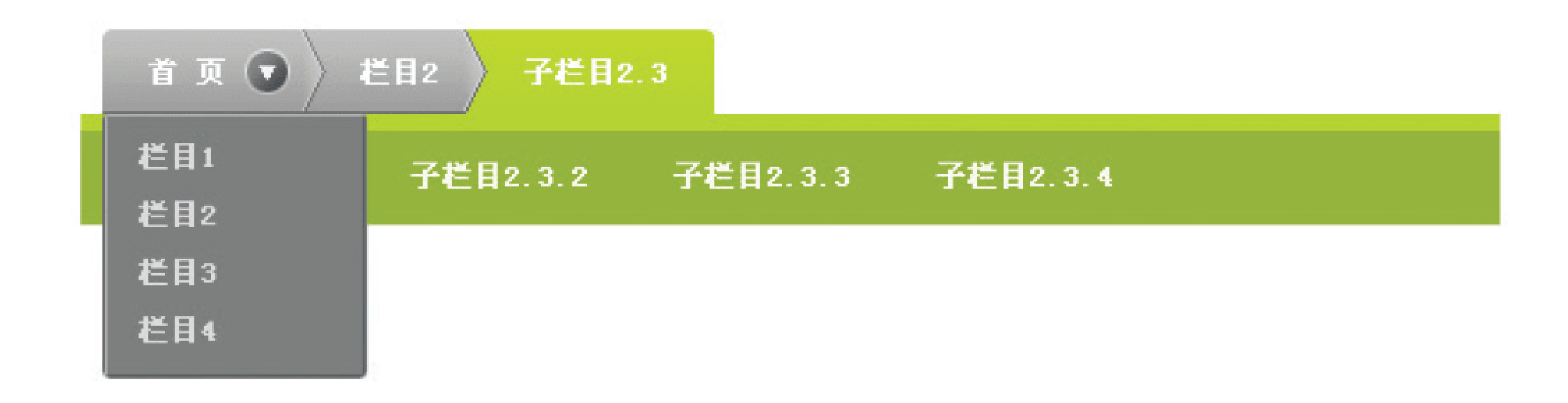


现在有些“面包屑”已经不再单纯只作为来路的显示。有时候我们也利用它能表现层级的特点，标识出能连到当前页面的所有路径。这样可以吸引浏览者访问其他页面。例如“卓越亚马逊”就利用这种方式来表现书库的不同分类形式，便于购买者根据自己的需要前往不同的上层页面浏览。



现在又出现了一种新颖的“面包屑”导航栏，它同时结合了标签导航的表现形式和功能，表达了网站的层次、结构，而且使用起来非常方便、一目了然，看起来很有新意。具体来说是如下一种方式：





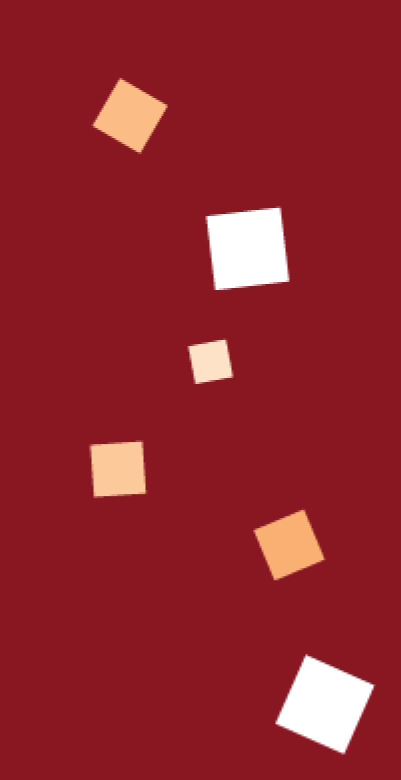
可以看到，当我们选择某个栏目的标签后，导航栏出现的是一种“面包屑”的形式。这样不管浏览者所处的层级有多深，他都能很清楚地看到现在处于哪个栏目之内。同时，他们能够使用上级标签的下拉菜单前往其他栏目。

Yahoo! 网站的很多版块就使用了这种“面包屑”导航。例如它的 Food、Sports 和 TV 版块。

总的说来，网页的可用性所需要注意的地方数不胜数，可能网页的任何一个角落都会出现问题，从而降低浏览者的好感度。我们知道了人们一般情况下是如何使用网页的，因此就需要针对他们的这些特点加以考虑。

平时在自己使用互联网的时候多加注意是一个很好的办法，这就好像是“大家来找茬”的游戏，只要看到那些让人郁闷、麻烦或者不愉快的设计时，就记录下来，和很多朋友或同事一起讨论，吸取它们的教训，做到“前车之鉴，后事之师”。

还是那句老话，如果希望自己设计的网站在推出后经得住用户的考验，就需要在设计期间经常地进行用户测试。测试能够帮我们及早地发现很多问题，从而避免以后的尴尬。



第七章

Web-based 产品



“微软推出Web-based Office应用令人联想到是为了确保地盘。你们看到Google及像Zoho这类新创公司抢先一步——而且还有在线EPS应用，微软却还慢调斯理。”

——CNET News.com访谈微软首席软件架构师Ray Ozzie

“把软件做成 Web-based 是 Web 3.0 的一个趋势”。

不管上面这句话是否正确，起码我相信 Web-based 产品是软件开发的一个趋势，而且已经逐渐呈现出势如洪流的发展特点。

这是因为在这个无时无刻都能“在线”的时代，如果拥有 Web-based 产品，我们便不用像桌面应用软件那样专门地购买或者下载应用程序，也不用把它们安装在自己的机器里。你所要做的，只是打开 Web 浏览器，然后就像登录自己的 blog 一样登入并开始工作就行了。

你自己的数据不用存在自己机器的硬盘上，不必再担心使用的是什么操作系统，是否空间不够，或者无法方便携带。那些数据都能像 E-mail、blog 那样存放在各个网站的服务器上，随时随地供你调用。

那么作为 Web-based 产品开发者的我们，又该注意哪些问题呢？

7.1 Web或者不是Web？——关于Web-based产品

Web-based 产品的载体是互联网，人们通过 Web 浏览器访问它。同样它也需要有自己的网址。此外，它也是通过超链接实现页面之间的跳转，使用表格来控制页面结构和显示，使用文本域、复选框和单选框来输入数据。

这些都是网页的特性。那么 Web-based 产品到底是不是网页？我们应该按照网页的设计方式来设计 Web-based 产品吗？或者 Web-based 产品是某种软件？如果是这样，用户界面方面又该注意哪些方面呢？

可以说，Web-based 产品属于网页的范畴，但它们又超出了网页的范畴。它们似乎是网页，因为从操作方式和外观表现的角度来说它们就是网页；但同时 Web-based 产品又不太像网页，因为用户并不会“扫描”这些页面。它们有些类似应用程序，更强调目的性和功能性。

有一些比较偏向于软件

用户上网浏览一般的网页是为了查找信息、阅读新闻，或者找点什么消遣一下。但如果他们上网是在使用 Web-based 产品，很明显那就是为了工作，或者完成某项具体的任务，例如在线编辑视频、使用电子地图查找行车路线。

有一些 Web-based 产品的功能性非常强，最常见的是那些网络管理系统。我曾经参与过很多这种类型的产品界面设计。有时候这些产品的界面完全就好像某些桌面应用程序。



这类 Web-based 产品其实就是某种软件，只不过它的载体是内部的网络平台。在进行这种产品的用户界面设计时，你更多地需要考虑其软件特性方面的可用性，例如界面如何安排得更为紧凑、元素间的层级关系等等。

有一些则更偏向于网页

但并不是所有的 Web-based 产品都像上面那样如同桌面应用程序般的冷静和精细。有些企业希望自己的系统能有更人性化的用户界面，希望员工更有归属感，使用系统的时候更轻松，而不是“被迫”地工作。

这种时候如果把产品设计得更像一个门户网站，使用更多的图片和轻松诙谐的文字，也许会减弱一些“工作”、“流程”等概念给人带来的反感。例如下面这个知识共享平台：



有些产品比较单纯

并不是所有的 Web-based 产品都那么复杂，有那么多强大的功能。有一些就好比是只有几百 K 的小程序，我们只使用它们做一些简单的操作，完成一些简单的任务。

如果任务比较单纯，就不要复杂化。简单的设计风格，清晰的信息结构都非常有利于用户的使用，他们很清楚：我只要这么一下，任务就完成了。别给他们太多不必要的压力。



对于这种 Web-based 产品来说，我们只用去关注任务本身，也要让用户只去关注任务本身。界面越简单清新越好。图形部分的设计就好比是味精，如果在烹调的过程中搁太多，只会令人反胃。

有些产品的用户群比较广大

还有很多的 Web-based 产品面对的是所有大众，例如那些全民级的视频网站或者 blog 系统，或者类似"Google 文件"那样的文档在线编辑服务。任何人都有可能登录网站，有些只是浏览或者阅读，有些则会上传自己的视频或者撰写文字让服务器存储。

正因为面向的是所有用户，所以我们不知道他们都是些什么人——智商如何、是否有耐心。我们也不清楚提供的服务和选项他们是否都了解，是否能适当地操作。

为了最大限度地保证绝大部分用户都能很快地学会如何操作，不管是上传视频、编辑视频、修改图片，或者是转载、引用博客、插入图片，我们都需要注重易学性，让用户更容易上手。即使

不太容易上手，我们也需要多动脑筋，怎样才能让他们更愿意学习、以及如何才能更轻松地学习。这就好比是设计“傻瓜”相机那样——充分为用户考虑。



最后……它们还是网页

不管怎么说，我们所做的一切都是为了用户。用户们认为绝大部分的 Web-based 产品仍然属于网页，因为他们需要通过 Web 浏览器访问网络才能使用这些产品。而且他们看到的同样都是导航栏、首页和表格，最多的操作也是填写各种表单。而这些在普通网站中同样经常看到（例如在注册新用户时）。即使有些 Web-based 产品非常像桌面应用程序，但他们也只会觉得这个网页的界面更“精致”和“紧凑”而已，而且这种类型的 Web-based 产品数量并不占很大比例。

人们对使用的是网页还是 Web-based 产品本就没有什么很清晰的概念，我们也没有必要去时刻提醒他们“你现在是在一个网站里面”或者“你正在使用我们的软件，它是基于网页形式的”。这对他们达到自己的目标没有任何帮助。也许还有很多人根本不知道 Windows 的桌面应用程序和网站之间的区别，但他们同样用得很好。


管它们是什么，我总得用鼠标和键盘。

另一方面，对于设计人员来说，Web 和 Web-based 产品的开发工具也很相似。这里所指的是页面的实现工具，例如 Dreamweaver，而不是具体的数据处理程序。同样的，在设计过程中使用的原型制作工具也几乎一样，例如我们都可能会用到 HTML 和 JavaScript。

也许你会说用户一旦在使用 Web-based 产品时，他们会更为小心翼翼，很担心是否会由于错误的操作而造成不好的后果。但是，Web 也好，Web-based 产品也好，你同样都需要强调用户操作和数据存储两方面的安全性。让用户更为轻松、放心地执行任务，提高他们的工作效率，这本来就是任何产品界面设计所要达到的目标。

7.2 我为什么说要考虑用户的感受

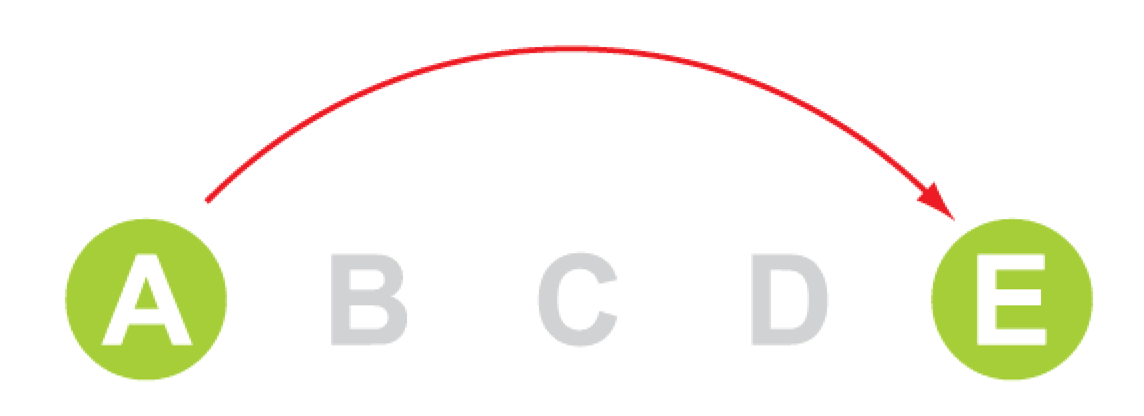
不管是 Web 还是 Web-based 产品，人们仅仅只是使用它们达成目标或者完成任务。他们不一定理解它们的运作原理，甚至对这些原理的理解是完全错误的。他们也不关心这些“背后的东西”。

这就是为什么“产品说明书”总是没人想看的原因。不管是冰箱还是 DVD 机，我们面对那些开关和按钮总是愿意“先尝试一下”，看看这样是否能行得通。我们不愿意去翻那些厚厚的、满是专业词汇的使用说明。

在使用互联网的时候同样如此。想想看，当你尝试想要上传自己的视频时，你会先去找“如何上传视频”的帮助链接么？我想可能 99% 的人会直接点击“我要上传”的按钮。

对于大多数人来说，是否明白事物背后的工作机制并不重要。他们关心的只是“我能不能正常使用它”。这不是智力低下的表现，而是我们并不关心——它没那么重要。

从这个角度来说，用户对产品如何运行并不感兴趣。你也没有必要去表现出来，因为这样反而会增加他们使用产品时的困扰——这并不容易学习。对他们来说，网页也好，Web-based 产品也好，无非就是对象和操作这两方面。我进行了什么操作，就能得到什么结果，其他的，让它们去好了，这不是我要考虑的事情。



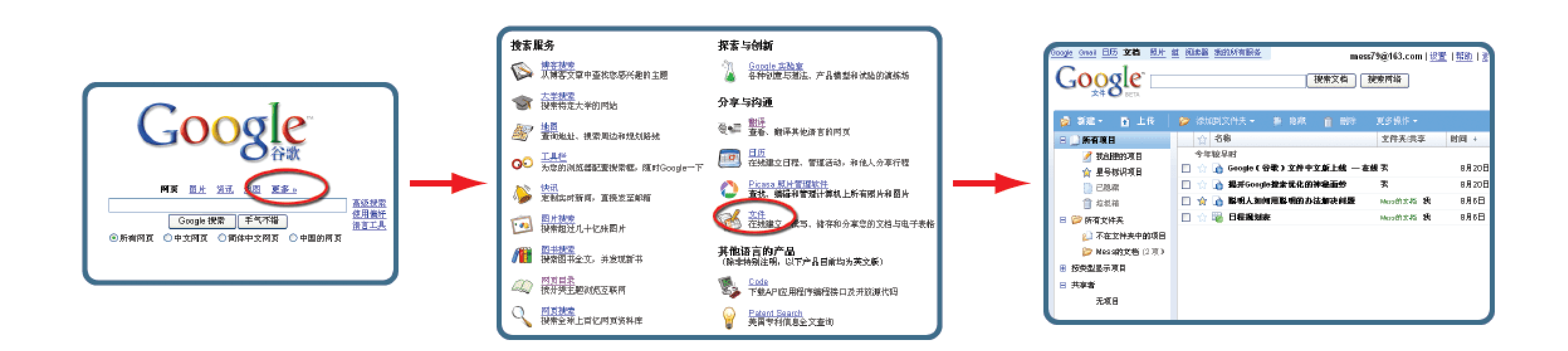
就像上面这幅图所表示的，如果理论上流程从 A 到 E 需要经过中间的 B、C、D 三个点，那么一般情况下，你呈现给用户的只是 A 和 E 就行了。用户不关心 B、C、D。

也就是说，网页或者软件的内部运转细节应当保留在内部，让用户看不见也想不到。

但另一方面，你仍然需要告诉他们怎样来使用产品。不是展现那些技术原理，而是如何才能更好地使用。这需要你仔细考虑如何去规划和展现用户界面。

人们在第一次接触产品后学会的方法，将会对他们日后的使用产生非常深刻的影响。当发现能够通过某种方式来使用某个东西（不管是电器还是软件），他们就会一直使用这种方法，哪怕它非常麻烦。他们不会去想“这样做是不是正确？会不会有更好、更方便的方法？”。也许当他们偶然发现一些更好的方法时，会改变自己的使用习惯，但仍然很少会去主动寻找捷径。

我的一位朋友第一次使用“Google 文件”时，从 Google 首页点击“更多”链接进入了“更多产品”页面，然后在里面找到了入口。



从此，他一直都是通过这种方式来进入“Google 文件”。他也想过这样好像有些麻烦，但没有人告诉他该怎么做更好。其实一个很简洁的办法就是输入这个产品的域名“docs.google.com”。



也许 Google 的设计师们看到这位朋友居然一直通过这样的途径进入他们设计的产品，会很有些哭笑不得。但人们总是跟他一样的思维模式：“只要这样能用，我干嘛还要去找其他方式？”

不管途径是什么，用户只想达到他们的目的。这就像有些人一直用钥匙掏耳朵一样，并不是说他们买不起掏耳勺。

然而，尽管用户毫不关心什么才是最正确的途径，你仍然得尽量去告诉他们：也许你这样是没错，但不太好。让我来告诉你更好的方法是什么。

这不是说你只需放一些“指示说明”的链接，然后提供给他们大段的技术性文字就行了。这只是一种敷衍的方式。“为用户着想”体现在一开始就提供最简明的操作界面、最合适的选项，同时提供耐心而灵活的提示和帮助。这样能够让他们轻松地、兴味盎然地以最好的途径来完成任务。

也许你会问：“既然人们总是习惯用自己的方式解决问题，我有必要这么麻烦么？”

答案是，很有必要。因为有时候他们自己的方式效率并不高，而且容易出错。

而且，如果用户明白了怎样做才是最好的方法，他们才能更容易、更方便地达到自己的目标。这会使他们更加信任你的产品。同时他们也会更容易地理解你的产品。另外，在明白了什么是最有效率的方法之后，他们便会觉得自己在这里更加聪明，更加游刃有余，这样能使他们成为更忠实的用户。

如果不去考虑这些，认为“用户能做就行了，没必要去操心他们到底是怎么做的”，那么可能导致的结果就是：一旦用户们偶然接触了另外一个网站或其他人提供的产品，而他们突然发现在那里能够更有效地完成任务……你觉得他们还会再回来么？

至少，当我的朋友发现“Google 文件”可以通过直接输入网址的方式进入之后，他有些埋怨他们：“为什么不在某个地方直接告诉我这项服务的域名？害我老是这么麻烦！”

7.3　首先看看菜单和对话框

Web-based 产品兼具“网页”和“软件”两方面的特点。在上一章我们已经专门谈过了网页方面的可用性，现在可以把注意力主要集中在其“软件”这一部分的特性上。

几乎每个 Web-based 软件产品都会有菜单（要不就是类似菜单的导航栏）。而只要用户进行操作，不管是正确还是错误，也都可能会出现对话框。我们就从这里开始。



"Google文件"的菜单和弹出窗口

7.3.1　我需要菜单，但必须是好菜单

对于大多数软件来说，菜单显示了大部分的命令和选项。我们曾经讨论过菜单的设计，例如不要把一些对立的命令放置在一起，以及适当地进行分组。但有时候设计人员也会犯一些其他类型的错误。

时隐时现的菜单项

很多设计人员经常犯的一个错误就是，试图随着程序状态的变化增加或删除菜单项，以减少菜单栏的大小和复杂程度。也就是说，当我选择了某个数据对象后，菜单只显示与这个对象相关的命令。它显示的选项依赖于我当前的活动。



当我写邮件时的编辑菜单



当我编辑附件时的编辑菜单

这看上去像是一个好主意，但实际上不是。它会把用户弄糊涂：我在不同的时候查看菜单，会找到不同的菜单项。如果用户只是刚刚接触这款产品，不了解其中的依赖关系，他们就不会明白为什么有些选项在某个时候存在，而在其他时候却消失了："那个编辑表格的命令跑到哪里去了，我明明在哪里看到过它？"

毫无疑问，这种设计的想法是好的：他们想通过限制在不同情况下的可用命令来尽力帮助用户——我只给你可以点击的，这样你就不会犯错了。他们假设用户在一开始就了解产品，知道什么命令对应什么对象。但大部分时候并不像他们想的那样。

增加和移除菜单项对用户来说并不友好，菜单里面的内容应该是固定的。你可以通过把不可用的命令置灰来达到禁用的目的，这样用户会容易接受得多。而且他们通过查看菜单能够了解大概有些什么命令，以及各种命令的大概位置，这样可以更快地熟悉产品。



当我写邮件时的编辑菜单

当我编辑附件时的编辑菜单

当然，如果用户的操作环境发生了变化，对整个菜单进行增加和删除是没问题的。因为这种改变不像之前只是菜单内部的变化，用户可以很明显地看到发生了什么。例如下面的菜单 ：



网易邮箱中“电子邮箱”和“通讯录”中的菜单

探头探脑的菜单项

有些时候，设计人员会把一个命令放到多个菜单里面去。



两个地方都能建新邮件



两个地方都能设置选项

这种情况发生的原因可能是设计人员不知道这个命令到底属于哪个菜单组，所以就每个组里面都放一个——不管用户自己是怎么分组的，他们都能找到它。这好像有些道理，但其实会带来更大的副作用：用户看见重复的命令，他们会想知道它们是否不同。而更糟的是，他们一般都会认为这些命令是不同的。

同时，这种做法会带来另一个问题：误导用户，让他们以为产品比实际上更复杂。很多人在初次接触某个产品的时候总是会先快速浏览一遍菜单，看看里面都有些什么，从而得到对产品大概功能的一个总体印象。放上其实是同样功能的多个选项会增加菜单中命令的数量，而这会增加它的“规模”。

在网页中也有这种类似的情况：很多看上去不同的超链接，例如导航栏中的选项、一张图片或者一段话里面的某个词，指向的其实都是同一个页面。也许这种想法来自于为了帮助用户发现信息——我不能肯定你能注意到这个链接，所以在其他地方再放上几个。但是这样做会让网站看起来比它的实际规模更大、更复杂，用户会更容易迷失。我说过，网站越简单越好，这样便于用户发现自己所处的位置。

7.3.2　让我陷入困境的对话框

对话框是什么？对话框是弹出窗口，但不是所有的弹出窗口都是对话框。当你打开某个网站时，可能会弹出一些广告窗口，但那些不是对话框。对话框是人机交流的一种方式：我在对话框里进行设置，计算机就会执行相应的命令。

对于对话框来说，首先要记住的一句话就是：对话框就是对话框，不要在里面放不该放的东西。这些不该放的东西包括菜单栏（或者导航栏）和最小化按钮。

不要在对话框里放菜单或者导航栏很好解释，因为如果出现了这些东西，用户就会搞不清到底哪一个才是主要的窗口，是应该在这个对话框里面继续操作还是应该回到主页面。

而经常发生混淆的地方在对话框的最小化按钮上。有些设计师让对话框也能最小化，或者缩小成一个任务栏按钮，但这是错误的。因为对话框出现的原因在于它在要求用户作出决定，例如是“确定”还是“取消”，如果没有确认的动作，用户将不能进行后续的操作。它不能被最小化后搁在一旁。

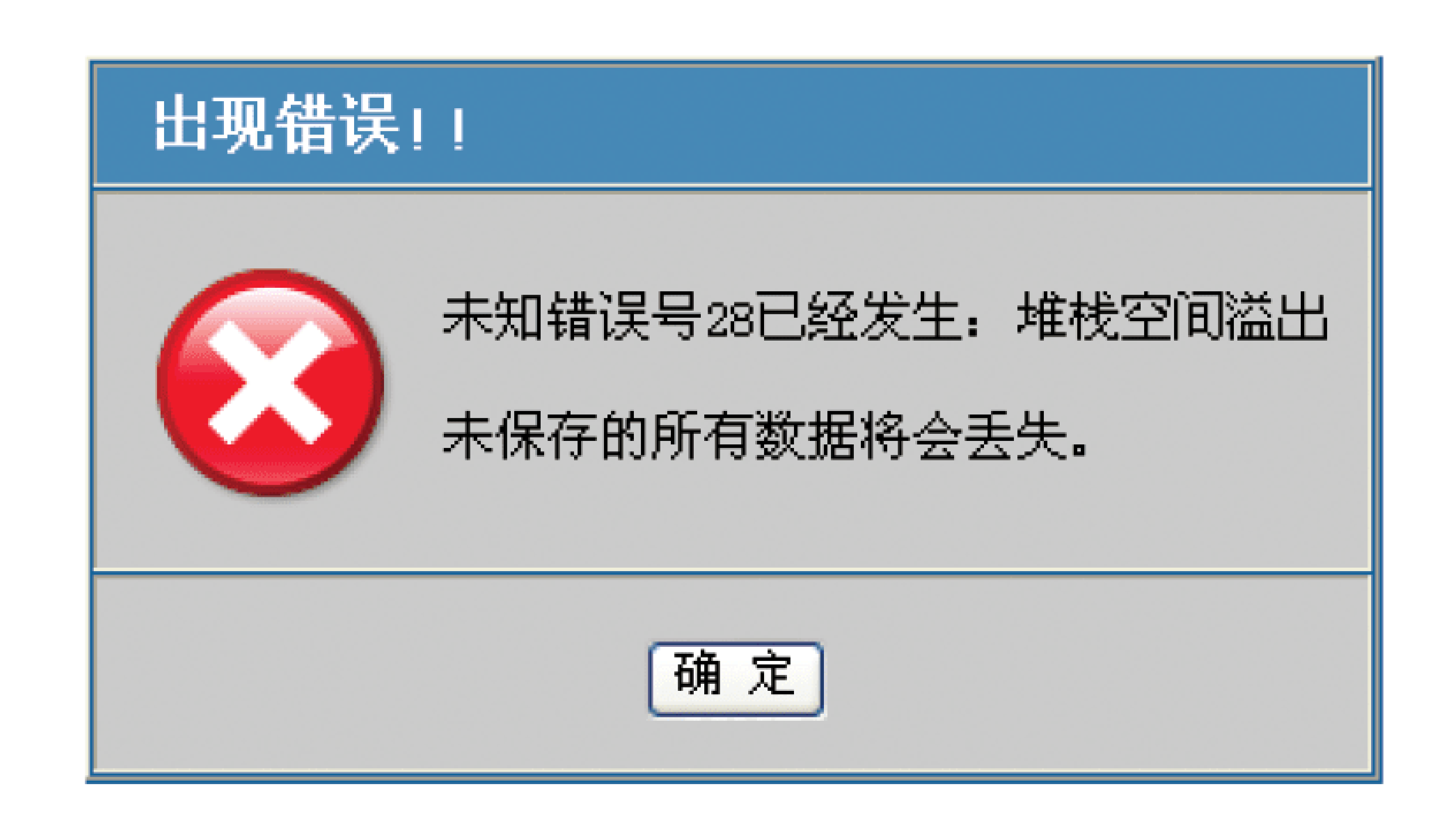
当出现了对话框时，如果用户把这个程序或页面的窗口最小化，对话框也会自动隐藏；而当用户随后恢复窗口时，对话框就再次出现。只有当用户明确给予对话框命令之后，它才会消失，任务得以继续。

此外，对话框同样也没有最大化按钮，也不能让用户改变它的形状大小。

但即使注意了这些问题，有些对话框还是会让人陷入困境。

让我没有选择

有些对话框显示一条信息，宣布某些可怕的事情已经发生或者即将发生，并且要求用户接受。即使用户能够理解问题的原因，但是除了叹气甚至咒骂之外他什么都做不了——只能确定这条消息，然后一切从头再来。



即使这种情况是无法避免的，但给这种对话框加上一个“确认”按钮只能是往用户的伤口上撒盐——都这样了，你还要我确认个什么劲？我一点也不想确认。

这让我想到一个笑话：“我有一个好消息和一个坏消息。坏消息是我没有好消息。”

让我不知道该怎么选择

有些对话框的用语过于含糊，而且存在着逻辑上的“陷阱”。

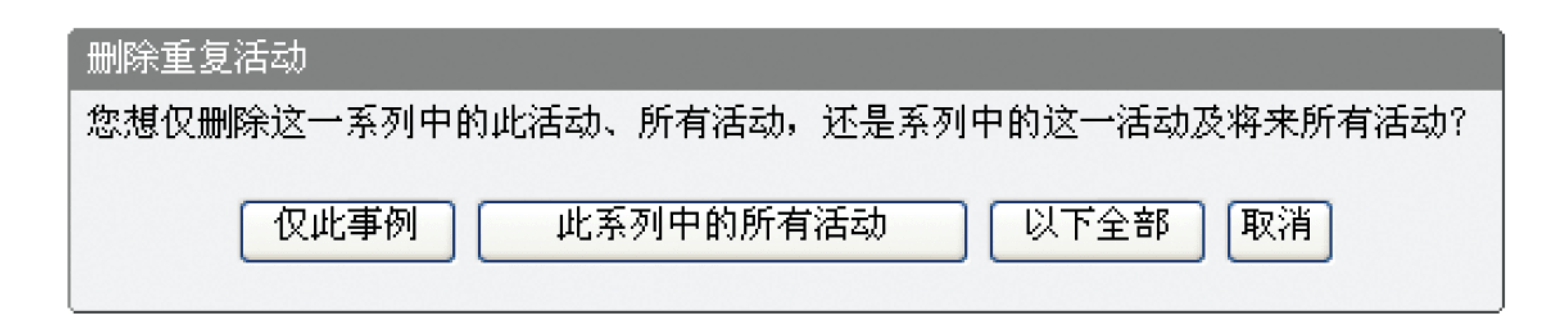
我在Gmail邮箱做了一些设置，但是忘记了保存，然后想写封新邮件。这时系统弹出这样一个对话框：



它的意思是问我是否保存这些更改。但上面的文字是“请问是否取消”。可能我不想取消更改。但是你知道我是怎么想的吗？点击“确定”表示“没错，我‘确定’我要取消”，点击“取消”似乎也表示“没错，我要取消，所以点‘取消’按钮”。似乎不管我怎么选择都会取消之前的更改。

遇到这种问题，用户只能对着这个对话框看一段时间，反复地咀嚼文字的含义，直到让自己相信某个命令是自己想要的。这种混淆可能会带来严重的后果。其实把按钮改成“是”和“不是”问题就能解决。

与此类似的，有些对话框把文字游戏玩得更加炉火纯青。这种时候用户好像正在参加“幸运52”的智力游戏。例如“Google 日历”中的一个对话框：



我既不明白它提出的问题是什么，也不知道它给我的几个答案该如何选择。

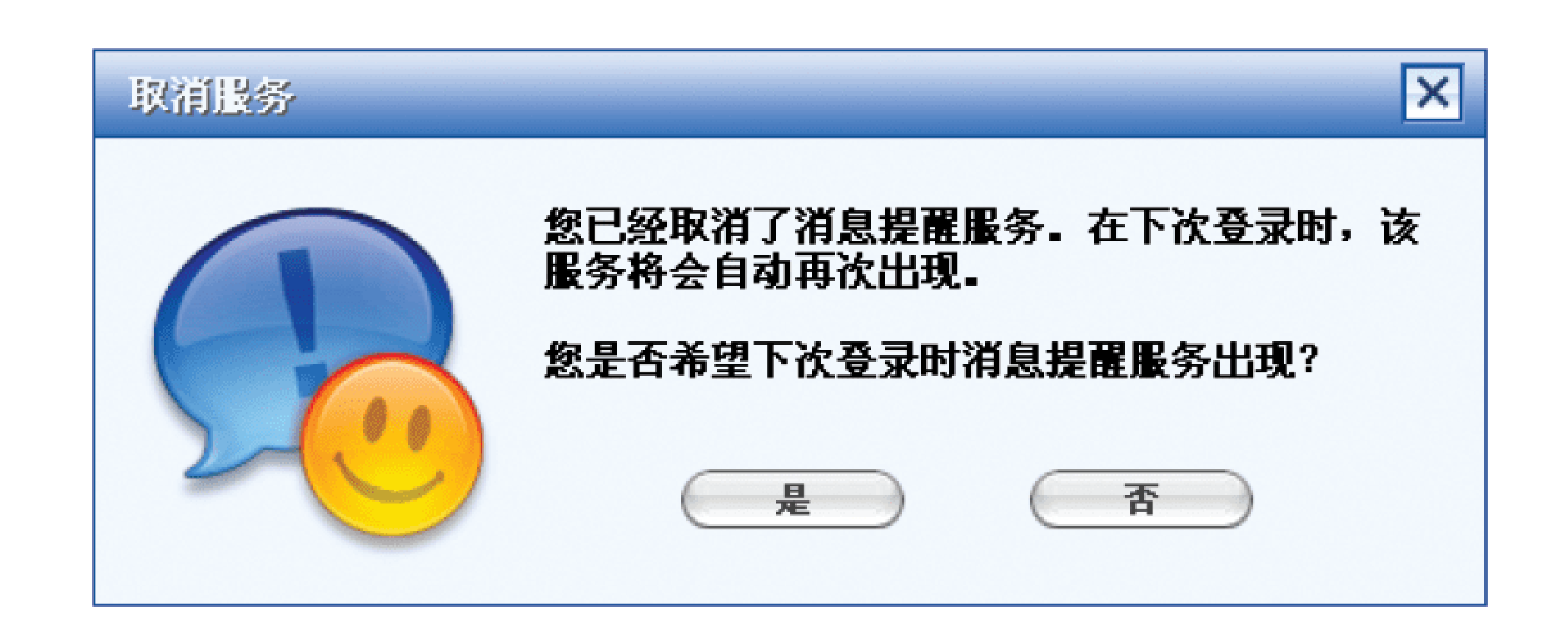
除此之外，还有些情况也会让你不知道该如何是好。例如有的对话框只告诉你一种可能的情况，而其他可能性根本不提。



程序说必须点“确定”来重启电脑。但这里还有一个“取消”键。如果我点的是“取消”会发生什么事情？程序没有告诉我。

没有第三种选择

另外一些不好的对话框给用户提供了多种选择，但却忽略了用户最有可能想要的那个。在某个网站中，我无意中点击了某个按钮，这时出现了下面这个对话框：



除了废话太多之外，这个对话框让我感到很沮丧的是：我并不想取消这项服务，这只是无意中的误操作。但我没有办法改变这一结果。我被迫在两个选项中选择一个："是，取消服务，但下次登录时让服务再次出现"和"不，取消服务，下次登录时也不要出现该服务"。没有第三个出口。我不能选择"唉，我不想取消这个服务"，其实这才是我最想要的选项。

7.4　你真明白那些控件的作用吗？

网页和 Web-based 系统都离不开控件。但让我吃惊的是，就连很多程序设计人员也经常会在这上面犯错误。

7.4.1　单选按钮和复选框

单选按钮叫做 RadioButton，而复选框叫做 CheckBox。

RadioButton 是"多选一"，这就好像收音机，你一次只能收听一个台的节目，要么就是中央人民广播电台，要么就是交通音乐台，要么是其他。而 CheckBox 则是"多选多"，就像你点菜的时候可以要一盘宫保鸡丁，一盘耗油生菜，然后再加个蘑菇汤。你选了其中一个也完全可以再选其他的，每个选项都是独立的，互不排斥。



单选按钮

启用下列事件的声音
新的系统通知　有新的邮件
故障告警　误操作警告
系统发生错误
确 定　取 消

复选框

但是就这样仍然还有人会出错。

把复选框当作单选按钮

最简单的例子就是把复选框的属性设为互斥。

消息设置
您希望对方在多长时间内回复本消息？
☐ 2小时内
☐ 一天之内
☑ 一周之内
☐ 无所谓
☐ 自己设定时间： 分钟 内
上一步 下一步 取消

用户只能够在这 5 项中选择一个。但是这种原本是可以选择多项的复选框会诱使他们尝试是否能多选，所以往往很多人都会在这种地方点击好几次鼠标：先选一个，然后再选一个，发现选了这个就不能选那个，于是再点回去。

把单选按钮当作复选框

有些地方只能用复选框。因为复选框可以表示“开关”的意思，而单选按钮则不行。但是有些设计师没有理解它们在这里的区别。例如：



记住，单选按钮是“多选一”，而且是“必须选一”。而复选框则可以多选，也完全可以不选。这是两者的根本差别。

是必须要我选，还是不需要我选？

很多没有足够用户界面开发经验的设计人员经常会出现这样的问题，即不给单选按钮任何的初始值：

您希望送给她的礼物是： ○ 鲜花 ○ 巧克力 ○ 项链

下拉选项菜单里面也是如此：

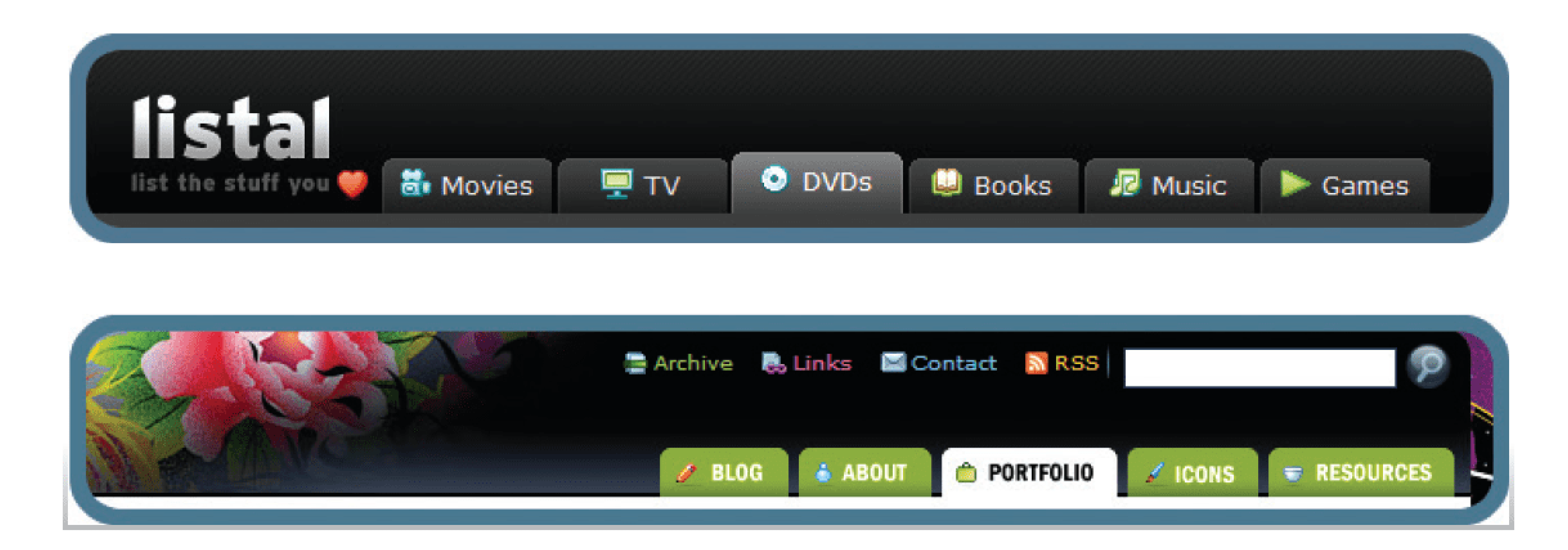


也许你不喜欢为用户事先指定某个选项，但是这样的表现方式会让用户感觉他可以什么也不选。单选按钮是“多选一”，它不应该处于无值状态，否则将会违反逻辑定义。在上面的例子中，也许用户不想送任何礼物，或者他只是不想送你显示的这三种。如果你允许用户可以有自己的选择，那么单选按钮就应该明确包括第四个值：“None”。

您希望送给她的礼物是： ○ 鲜花 ○ 巧克力 ○ 项链 ◉ 啥也不送或者其他礼物

7.4.2 标签虽好，问题多多

标签是一个非常棒的用户界面元素。不管是作为操作系统控件还是 Web 页面元素，使用标签都能得到很不错的效果。这是因为它的含义非常明显，体现出强烈的物理空间感——当前的标签在物理上位于最前面。





标签面板能在相对紧凑的空间内组织大量信息和控件，这是因为它给人一种遮盖的错觉：现在用不上的东西可以不显示，而当我需要的时候则随时能够像翻书一样地显示出来。

但这同时也造成了很多错误的使用。

标签只能用来导航，不能选择数据

有一个网络管理系统允许用户选择不同的方式来保存报表数据，这些方式包括日报表、周报表、月报表和年报表。设计师在这里使用了标签：



设计师希望用户首先点击标签页来选择需要的报表种类，然后对标签面板中的选项进行设置，最后点击对话框底部的“确定”按钮。

但是，很多用户抱怨说他们总是得不到正确的操作结果。有人说他设置了周报表，但存储的结果却变成了日报表或者月报表。

这是什么原因呢？原来存储报表的种类依赖于当用户点击“确定”按钮时，碰巧哪一个标签页被置于最前面。可能会发生这种事情，用户选择了周报表，进行了设置，然后点击了另一个标签，检查了一下选项，决定不在它上面做任何修改，接着点击“确定”。但这个时候其实当前的标签页已经不是周报表了，他做的设置已经被忽略了，或者早就消失了。

这就是设计师和用户对标签面板理解的不同造成的。用户认为标签起到的是导航功能，我在标签 A 里面做的设置，切换到 B 里面时 A 仍然存在。但设计师却认为你切换到了 B，表示你就要用 B 的种类来存储。

问题的关键在于标签不能当作单选按钮。也许这样改一下就不会再有误解了：

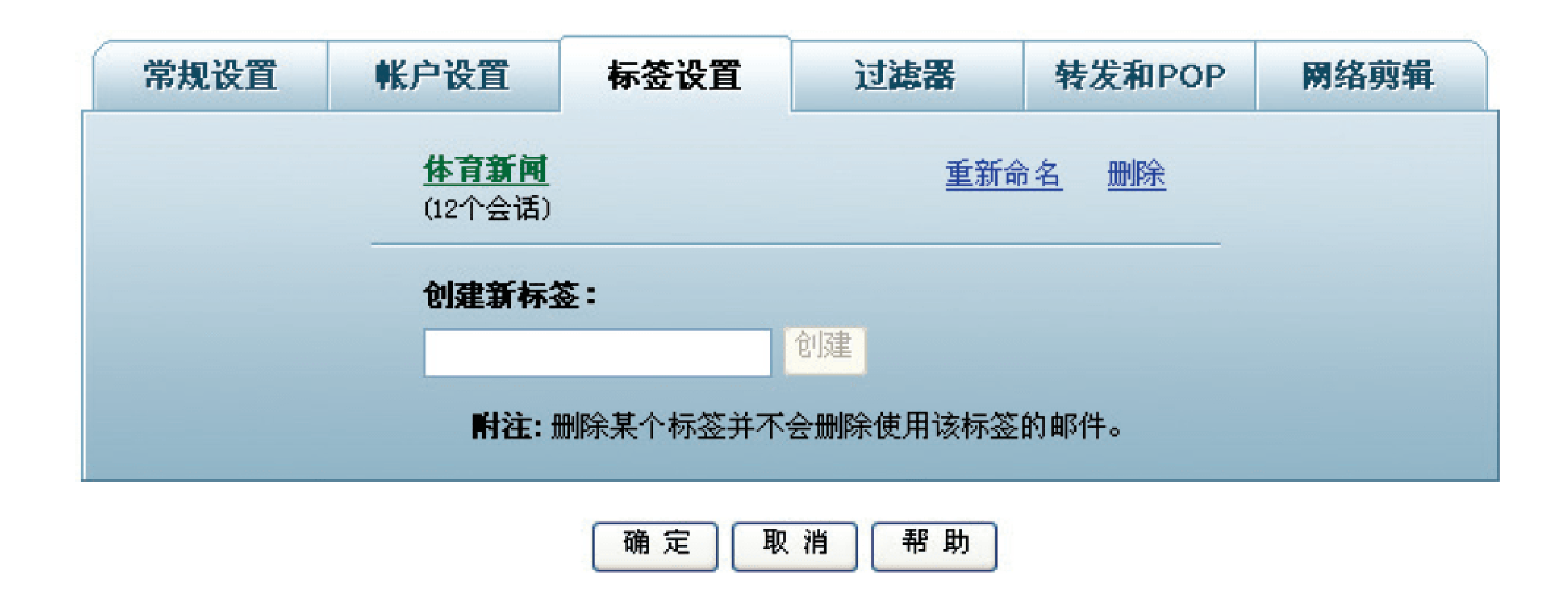


到处都是的标签

永远不要放置太多的标签。如果使用了过多的标签，你会发现只有以下几种办法来解决出现的问题，但是它们中的任何一个都不太好：

- **我把面板的宽度增加，这样就能放更多的标签**

增加了页面的宽度，确实能多放一到两个标签。但是不要忘记，每一个标签面板中一般都只有那么多的选项和数据需要显示，如果增加了宽度，每个面板中都会有大量的空间被浪费。



设置标签本来是为了节省空间，但现在却变成了你为了使用标签而故意去浪费空间。这有点“舍本逐末”的意思，不是吗？

- **顶部放不下了，就在其他地方放标签**

为了避免上面的情形，于是有些设计师在面板的左边或右边添加标签，甚至在下方。

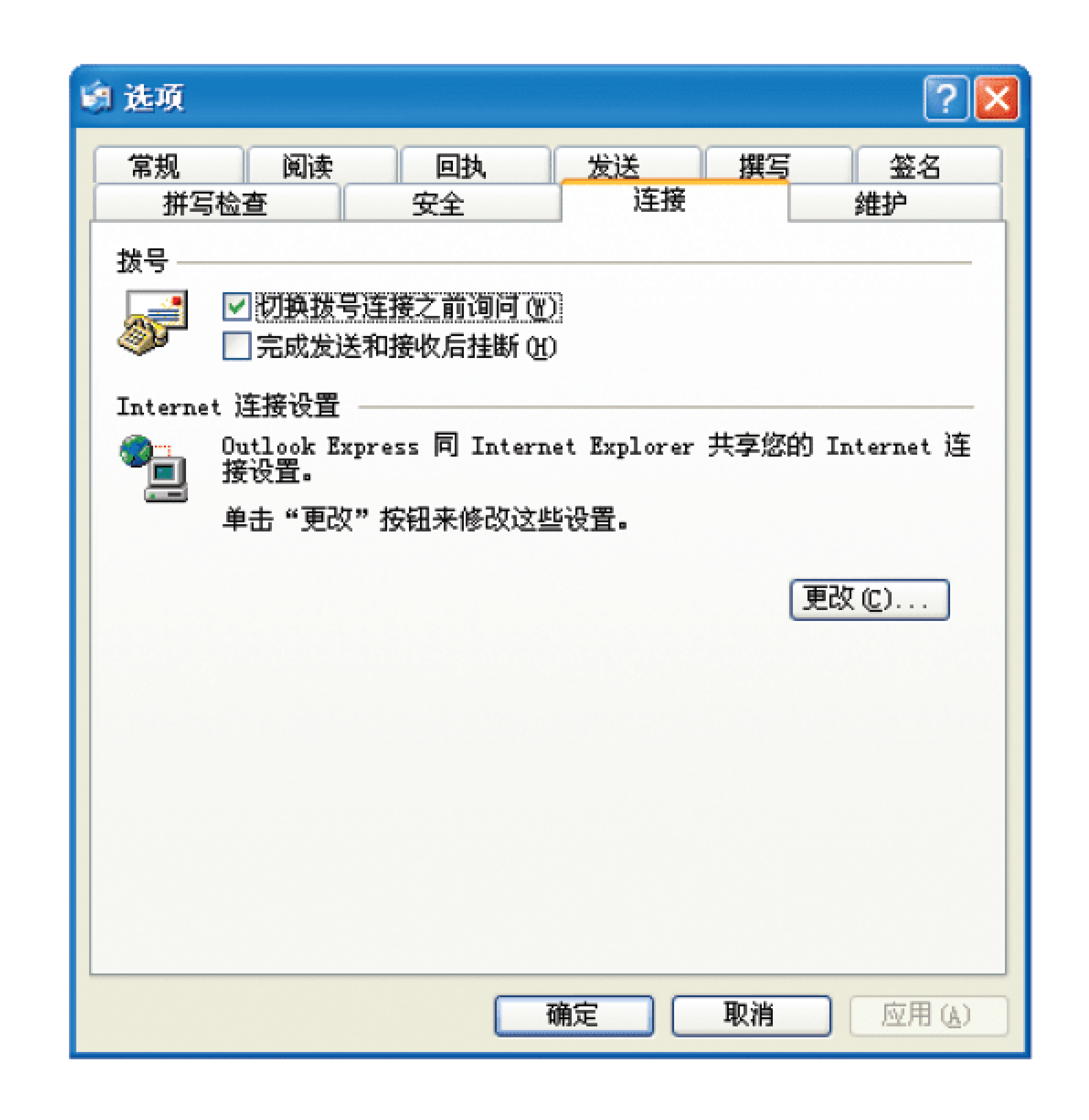


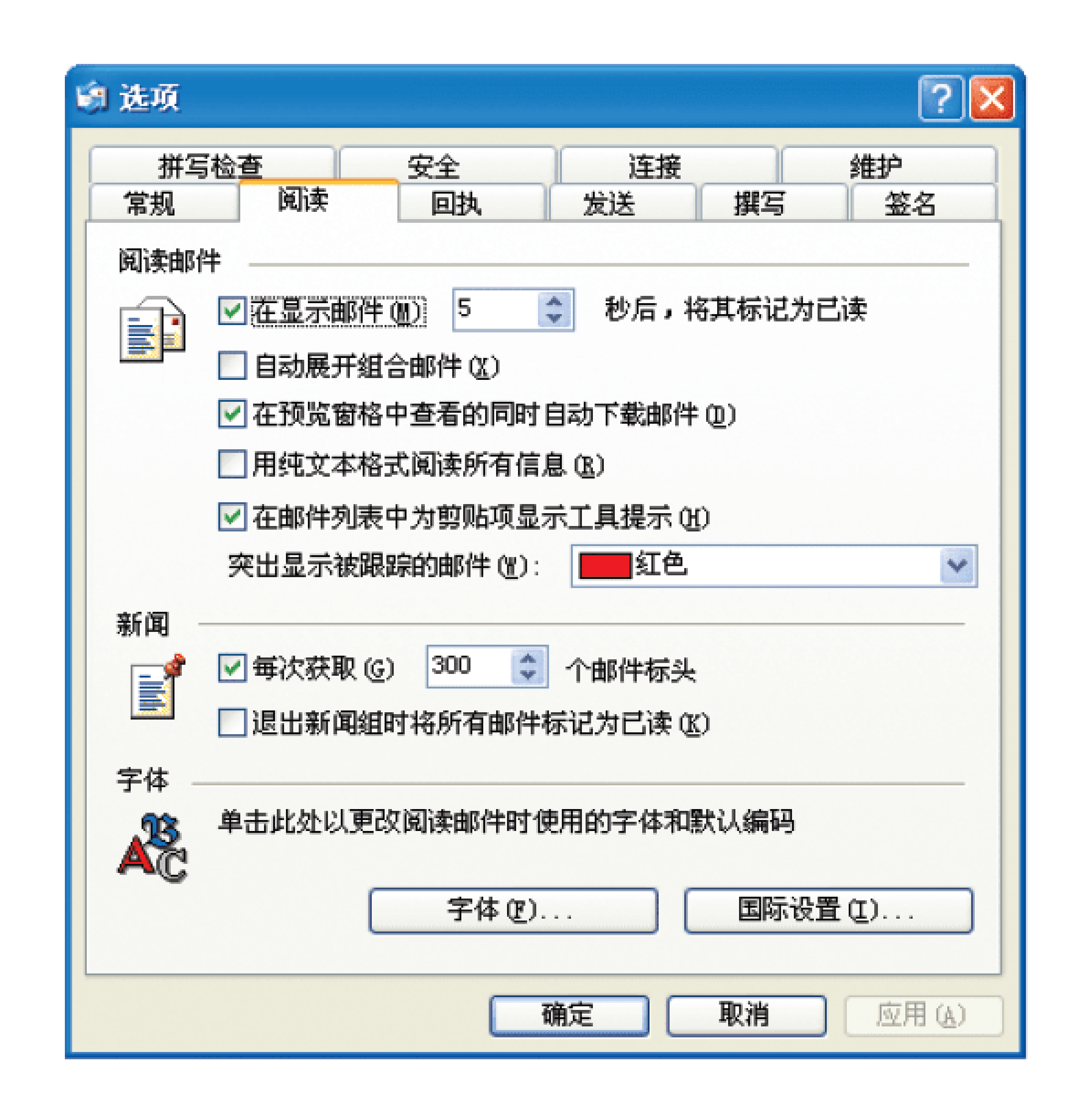
但是，这种方法不适合用户的使用习惯。他们会认为顶部的标签是一级分类，而侧边或下部的标签只是这些分类的附属选择。他们绝不会想到这些标签其实是统一等级的。而且，这种方式看起来令人很不适应——超级复杂的界面。

- **那我就放两层**

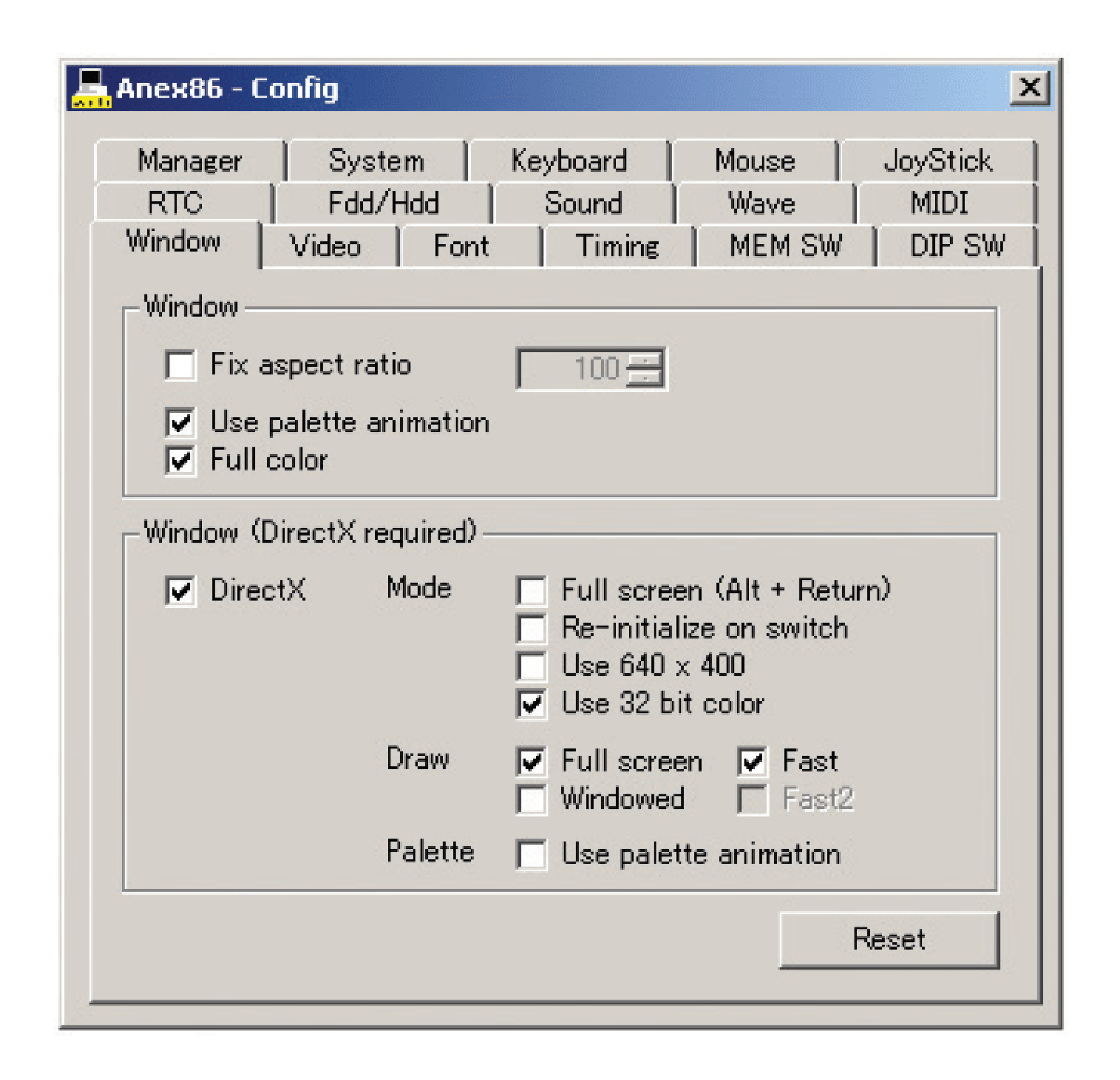
为了节省空间，同时又不能放在除了面板顶部的其他地方。于是一个很流行的解决方法是用两行或者三行标签来代替单行显示。这是个很好的主意，但也有不足之处。

由于标签的特点在于它把当前页面显示在物理的最前面，所以选中的标签必须和面板相连接。这就导致一个问题，就是当用户选中第二行的标签时，第二行会移到第一行来，并显示相应的面板。同时，之前的那一行标签变成了第二行。例如 Outlook Express 6 里面的“选项”对话框。





用户会发现，当他点击了上面一行的某个标签后，之前的那个标签好像消失了。其实这个标签只是移到了一个新的位置，但这可能要花费用户一些时间来找到它。当标签过多的时候尤其令人烦恼，例如下面模拟器 Anex86 的设置界面。



- **用标签嵌套标签**

还有一个常见的问题是在某个标签面板中设置了一组新的标签页。这会让用户感觉眼睛看不过来。

曾经有一位客户让我修改他们Web-based管理系统的所有界面。我发现其中一个页面中就嵌套了两组标签，而且惨不忍睹，我几乎都难以把它作为一个用户界面的例子：



其实，这里上层的“Named”和“Zone”完全可以用单选按钮来代替。在调整一下之后页面并不会这么难看：



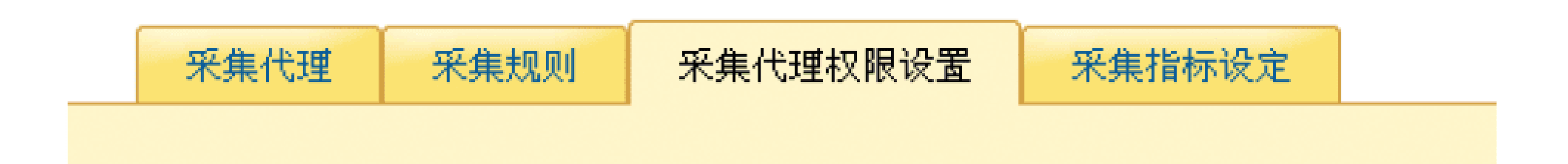
如果标签太多，说明两个问题：一、你的分组可能没有经过仔细考虑。二、你忘记了可以使用其他控件来代替标签。

太小和太大的标签

使用标签时还有一种错误的使用在于它的长度和高度。其实这种大小主要是由于里面使用的文字导致的。

- **有些标签页超长**

出现这种问题时大多是因为没有仔细考虑用词，导致有些标签太长。



其实后2个标签里面的“采集”两个字完全可以去掉，因为这4个标签都是针对“采集”而言的。

- **有些则太高**

而有些情况下似乎标签页的文字无法再缩减，尤其是对英文来说，单词长度是无法控制的。那么有些设计师会把一个标签分为两行来安排文字。



但如果有的标签是一行文字，有的又是两行，这样看上去会相当糟糕。

- **还有些则太短**

对这个问题也有解决的办法，就是使用英文缩写。但是切忌过分地缩写，否则就是在牺牲标签的清晰程度，甚至可能还会使用户出现误解。



“Info”代表“Information”没有问题，但其他几个缩写就不明所以了。

7.4.3 简单而又不简单的文本框

文本框是最常见的界面控件之一，因为绝大多数软件产品都需要用户输入数据，而不可能只让他们对现有数据进行选择。

也许你会说，这个玩意不就是个输入区域么，还有什么讨论的必要吗？答案是仍然有必要，因为我见过太多的设计人员在这上面犯错误。

这些错误主要包括：

在不该用的地方使用文本框

我经常在一些网站或产品中发现有些文本框不能编辑。它们看上去处于未激活状态，有时候里面的值也可以改变，但不是通过编辑文本框来改变的，往往是其他操作使它发生变化。这些文本框本身一直都不能编辑，其实它只能显示数据。

设备属性

设备名	HKJM-12008-1	别名	HKJM_12008_1zzzz	管理状态	正常
sysName	测试中文HKJM-1200	SNMP地址	221.11.129.33		
SNMP版本	2.2	sysDescr	Cisco Internetwork Operating System Software IOS (tm) GS Software (C12KPRP-K3P-M), Version 12.0(28)S6, RELEASE SOFTWARE (fc1) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2005 by cisco Systems, Inc. Compiled Tue 27-Dec-05 16		
sysObjectID	1.3.6.1.1.1.9.82	sysUpTime	104392443	sysServices	4
sysContact	ip@cnc-hn.com	sysLocation	HIHK-JM	ifNumber	11
SNMP读串	●●●●●●●●●●	SNMP写串	●●●●●●●●●	snmp端口	161

上面灰色的文本框其实在任何情况下都不能编辑，它们只是一个标记而已。这种时候，你需要把它们直接设置为简单的文本，而不是文本框。记住，不能编辑的文本就不要放在文本框内，或者是看起来像文本框的地方，因为文本框本身表示能被编辑，否则就是误导用户。

设备属性

设备名称		别名		管理状态	
sysName		SNMP地址		SNMP版本	
系统描述	Cisco Internetwork Operating System Software IOS (tm) GS Software (C12KPRP-K3P-M), Version 12.0(28)S6, RELEASE SOFTWARE (fc1) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2005 by cisco Systems, Inc. Compiled Tue 27-Dec-05 16				
sysObjectID	1.3.6.1.1.1.9.82	sysUpTime	104392443	sysServices	4
sysContact	ip@cnc-hn.com	sysLocation	HIHK-JM	ifNumber	11
SNMP读串	••••	SNMP写串	••••	snmp端口	
操作系统		软件		联系人	

我认为有必要归纳出计算机上显示文本的几个类型：

1. **居住地：** 北京市海淀区清华大学 可编辑的文本框

2. **居住地：** 北京市海淀区清华大学 未激活的文本框

3. **居住地：** 北京市海淀区清华大学 的文本

这是显示文本数据仅有的三种选择。

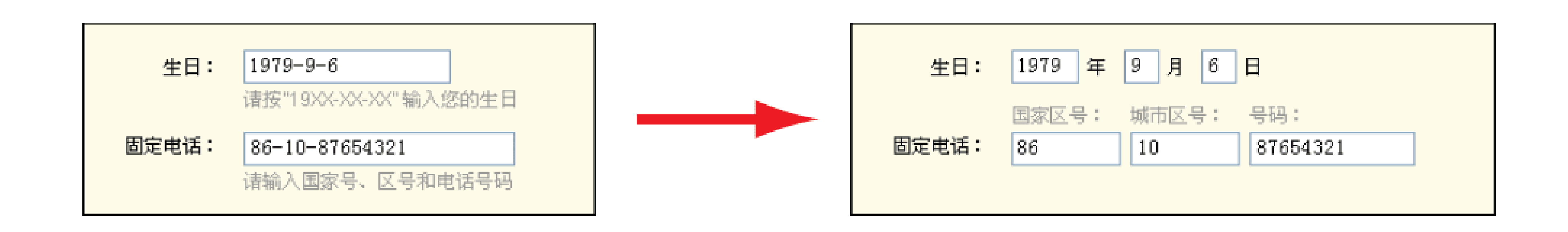
该用文本框的地方却又很随意

对于那些需要用户输入数据的文本框来说，也存在着一些可用性问题。

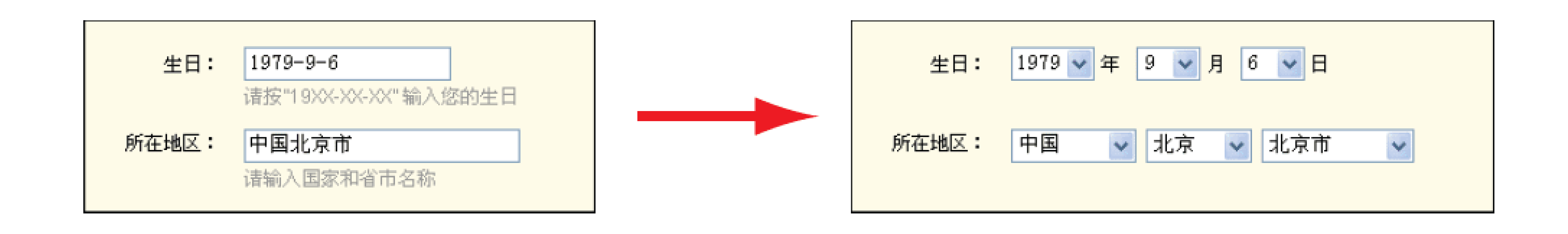
有时候我觉得文本框就像个来者不拒的垃圾箱：只要是文本，不管什么你都能输入进去。它给用户的提示太少，对于各种没有结构、格式自由的数据，它都能接受。但是，如果系统对输入数据有要求，它便会当你在文本框内输入之后才检查，而一旦数据不合法，就会提示错误信息。



有些设计师仅仅只是把文本框放在页面上就不再理会。用户如何输入是他们自己的事情，反正最后他们总会输对的——只是可能得费点脑筋而已。稍微好心一点的设计师会在文本框后面加上格式说明文字。例如“请在您的宅电号码前加上区号，并以‘-’间隔”。但更好的办法是多为用户考虑，把对应数据的各个部分使用多个文本框。这样用户就不需要考虑那些标点符号，出现语法错误的几率就会大大降低。例如：



而更进一步，你也可以使用专门为数据类型定制的控件，这样能匹配得更好，用户绝对不会输入无效值。例如：



7.5 每一个元素都要有合适的位置

一个产品就好像是一本杂志、一本书或者一部电影。它们同样需要初期的大量投入，然后让用户在与之交互的过程中获得各种体验。可惜的是，很多设计人员并不习惯把自己设想为产品的发行者。他们只是机械地开发，沉溺在功能和控件之中，从来不去想象日后用户见到产品时会有什么感觉。

从这个角度来说，一个好的用户界面，它的布局和外观也同样重要。你应该像那些出版商一样经常去关注产品外在表现的细节。

就像我的朋友 Scott 所说的：“尽管我只是个写旅游手册的，但也有作为作者的权利。如果出版商把封面设计得太烂的话，我绝对不会让它出现在货架上。”

7.5.1 控件的摆放位置

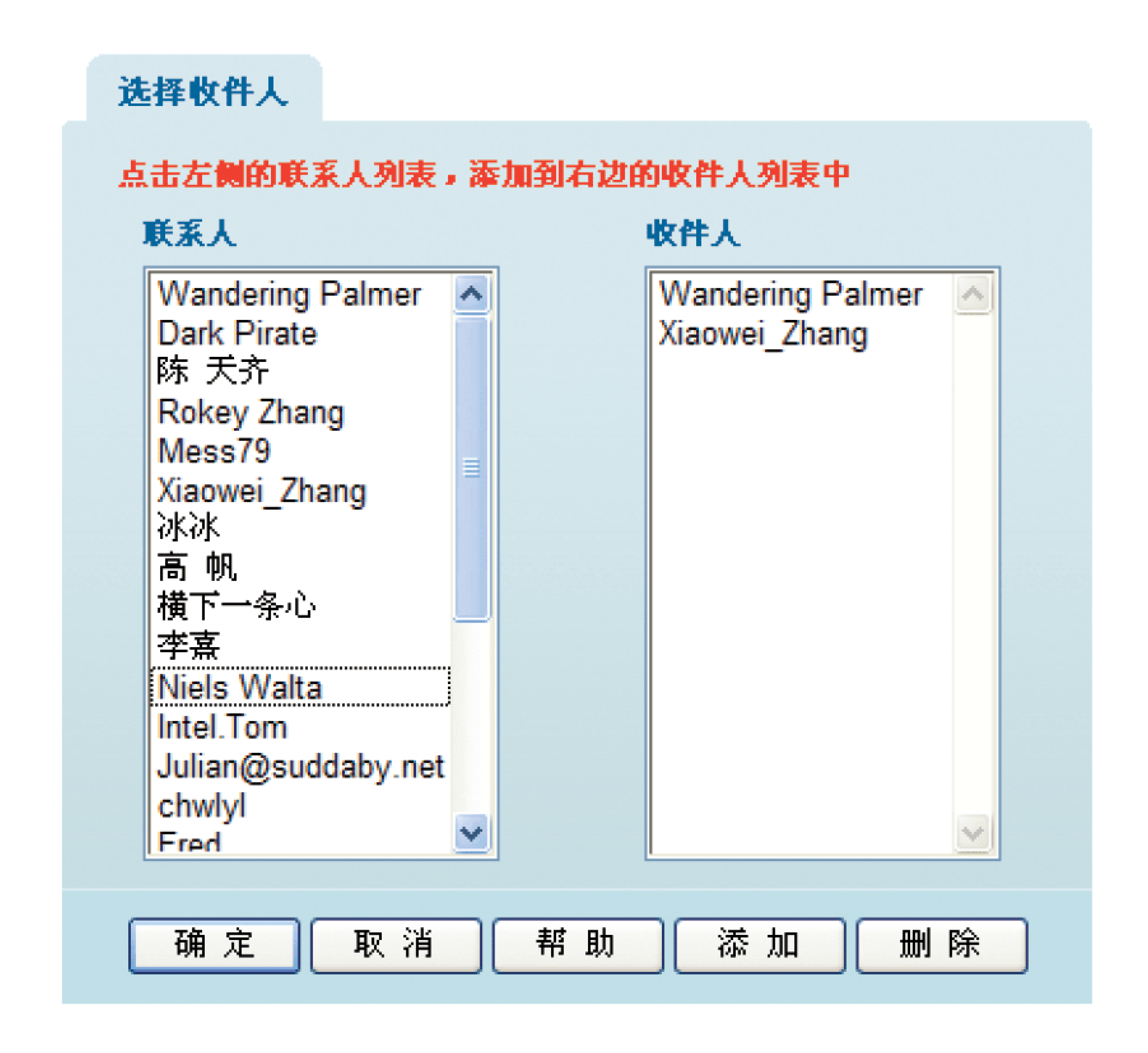
我发现有些设计人员，尤其是程序设计师，似乎并不在乎产品的外观表现。他们把全部的注

意力集中在功能上，实现了一个功能就把它放在一边，然后投入到下一个功能的实现中去。

这就导致一些控件或界面元素出现在不该出现的地方。

胡乱放置的按钮

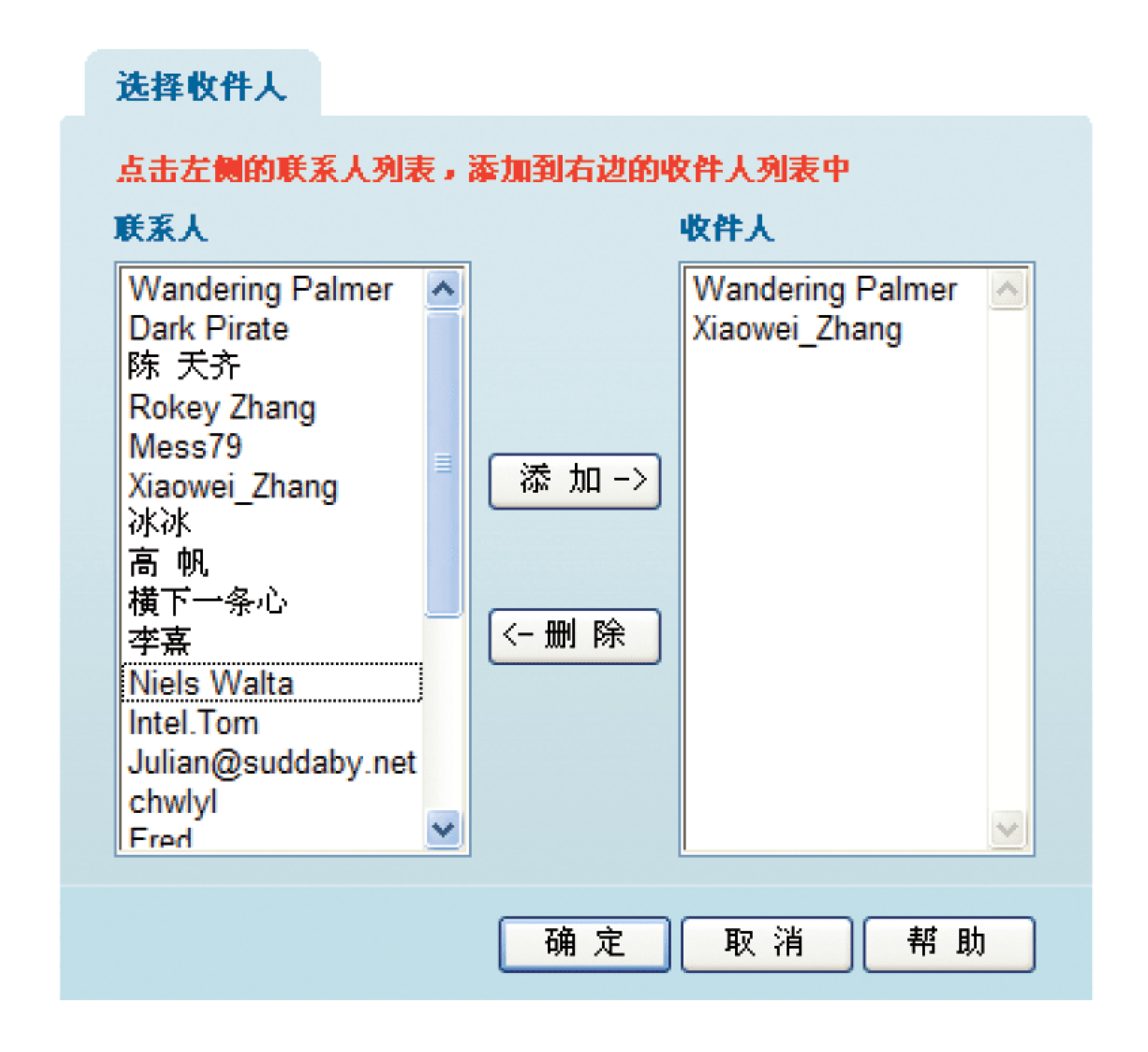
很常见的情况就是编程人员把两种按钮混为一谈，堆砌在一起。一种按钮是控制具体数据或设置的按钮，另一种则是标准的对话框控制按钮，例如“确定”、“取消”和“关闭”。



但实际上，这两种按钮所针对的对象是不同的，把他们放在一起会增加用户的困扰。也许编程人员认为所有的按钮都应该放在面板的底部，这样强调了一致性，或者他们认为这样有效地利用了对话框的空间。

但是，这种无谓的“一致性”带来的负面影响更大。用户在这个时候更希望的是操作数据，把控制数据的按钮“隐藏”在控制对话框的按钮之后，只能让他们花费时间来寻找。

正确的设计应该是突出控制左右列表的按钮：



距离产生美?

控件之间的距离也是个很重要的问题，太远不好，太近也不好。很多设计人员对这个方面考虑不多，或者也仅仅只是随意放几个空格了事，因为他们觉得这种距离的设置又花功夫，又没什么很显著的效果。

但是，也许考虑间距说不上能有多么好的效果，但是不考虑间距却很可能会极大地影响效果。

- **距离太远影响效果**

有些控件之间的间隔太远，以至于看上去根本就不像是在一个设置里面。



- **距离太近同样影响效果**

另外一些控件间隔过近，尤其是当很多的时候，用户会发现他们很难推断哪些控件属于哪个分组。这是由于当人视线水平横向移动时，很容易偏高或者偏低，尤其是当附近有其他元素时。

您的收入水平：	1000 元以下	2000-5000 元	5000 元以上
您的家人数量：	两口之家	3 人	4 人以上
您的出差频率：	很少	一般	经常
您是否经常与家人共度周末：	很少	一般	经常

这种情况下良好的布局就显得很重要了。要让用户能够顺利地完成任务，就得帮助他们不花太大功夫就能识别出各个控件是如何分组的。间距是一个很重要的因素，有时候稍微进行一些调整就能让局面有显著的改观。

- **加大不同组别的间距**

只要让不同组别之间的距离大于同组内控件的距离，人们就能很轻易地识别分组。

您的收入水平：	1000 元以下	2000-5000 元	5000 元以上
您的家人数量：	两口之家	3 人	4 人以上
您的出差频率：	很少	一般	经常
您是否经常与家人共度周末：	很少	一般	经常

- **把同组的控件安排得更紧凑一些**

如果各组的控件数目不一样，或者每一组的长度差别很大，这时没有必要去强调控件之间的对齐。让每个控件相距更紧密些，即使这样会打乱原本整齐的排列。

告警显示颜色：	亮绿	紫色	亮青	黄色	棕红	红色
事件类型：	邻居关系丢失	邻居关系发现	缺省路由丢失	缺省路由发现		
故障类型：	设备故障	链路故障				
故障状态：	当前	历史	全部	已修复		

这种不规则的布局能帮助用户利用视觉暗示来轻松地识别各个分组。哪怕组与组之间的距离小点也没关系。

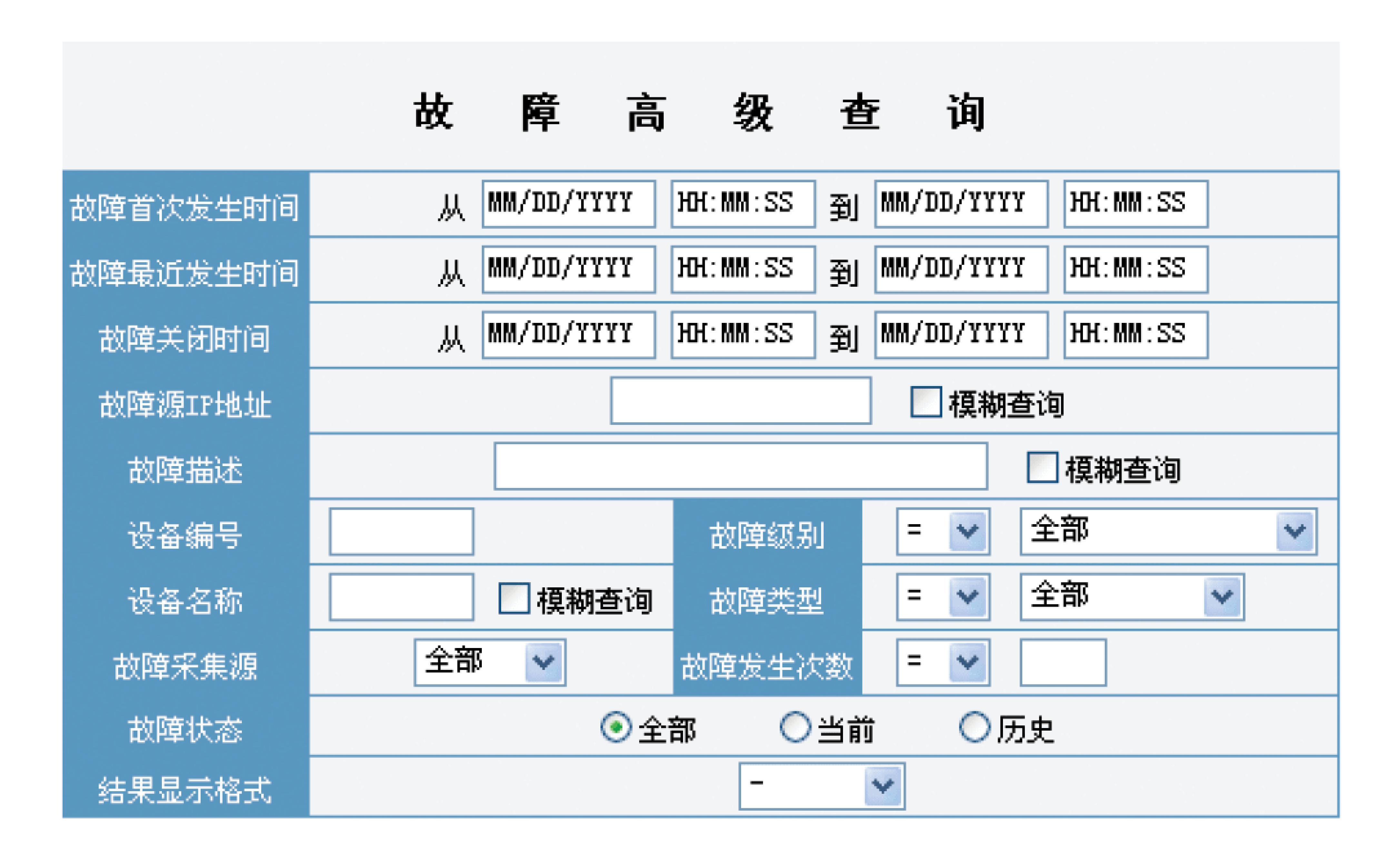
- **加一条分隔线**

如果你还是担心拉大分组间距离的方式不太有效，那么最简单的办法就是加一条分隔线。这样你还有疑问吗？



7.5.2 你无法逃避的对齐方式

不论是控件或文本，还是表格或图片，对齐都是我们最经常面对的问题。如果不考虑对齐的方式，或者只是随意设置，就很可能会造成可用性方面的问题。例如下面这种很糟糕的对齐：



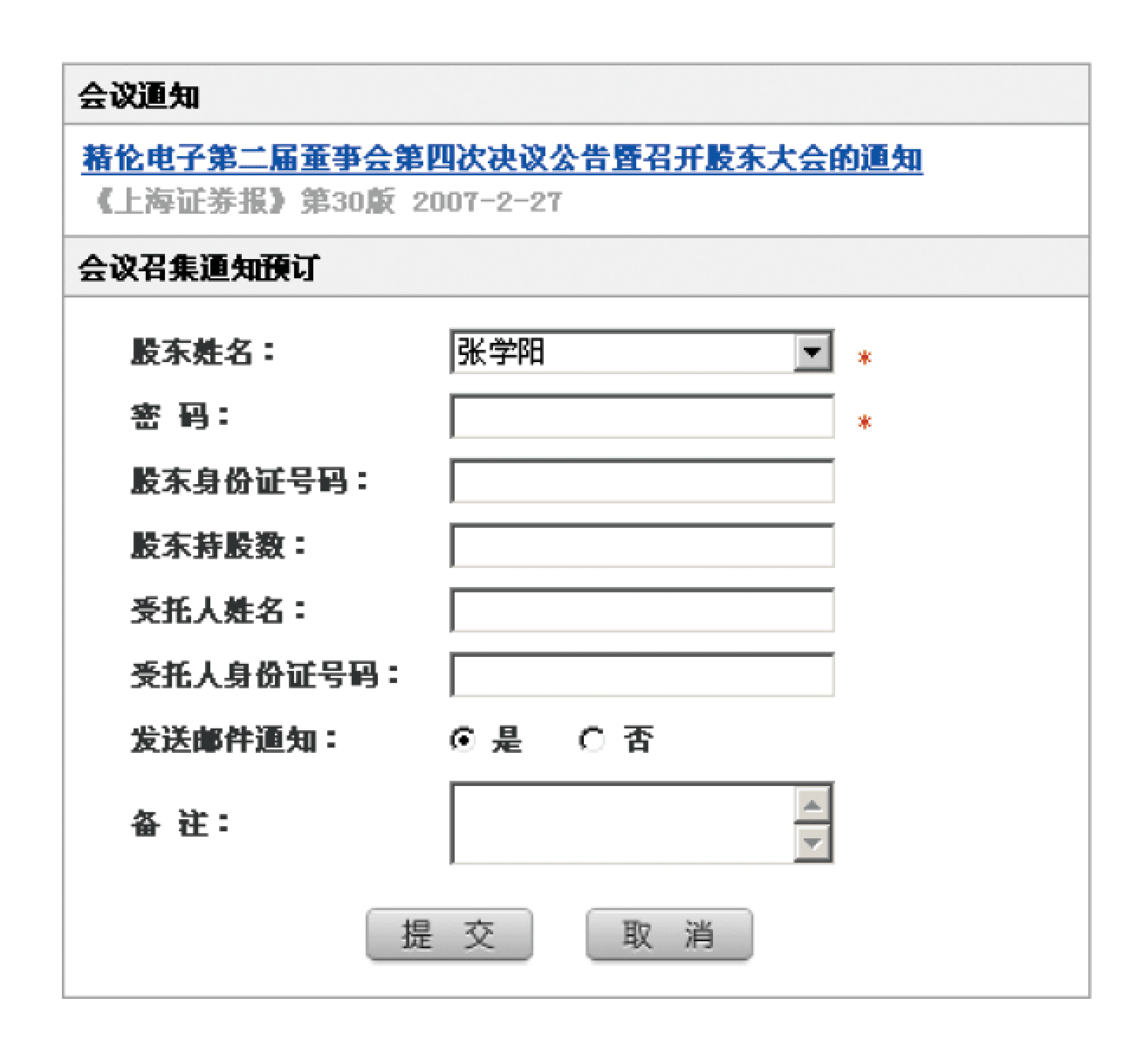
把对齐方式改一下就会好很多，用户看起来能一目了然：

故障查询
快速查询 高级查询
首次发生时间： 从 MM/DD/YYYY HH:MM:SS 到 MM/DD/YYYY HH:MM:SS
最近发生时间： 从 MM/DD/YYYY HH:MM:SS 到 MM/DD/YYYY HH:MM:SS
故障关闭时间： 从 MM/DD/YYYY HH:MM:SS 到 MM/DD/YYYY HH:MM:SS
故障源IP地址： 模糊查询
故障描述： 模糊查询
设备编号： 故障级别： = 全部
设备名称： 模糊查询 故障类型： = 全部
故障采集源： 全部 故障发生次数： =
故障状态： 全部 当前 历史
结果显示格式： -

是左对齐还是右对齐

对于绝大多数软件来说，例如 Dreamweaver、Flash 或者 Photoshop，一般默认的对齐方式都是左对齐。左对齐的排版方式最合适于习惯从左至右书写文字的国家和地区，因为人们浏览和输入的方式是一样的。

从垂直方向来说，左对齐的方式减少了用户眼睛移动和处理的时间。他们只需上下扫视一遍左侧的标记（Label），就能了解所要输入的所有内容。但是，由于有些标记可能很长，就会造成它们和对应的输入框之间的距离加大，这会影响用户填写表单的时间。他们必须左右来回地跳转目光来找到两个对应的标记和输入框。



于是一种替代方案产生了：设计师把标记进行右对齐，这样可以让它们和输入框之间的距离更接近，从而使得它们彼此间的对应关系也更清晰。但是这样也有不好的地方：左边出现了参差不齐的空白，用户们发现检索输入内容变得有点困难。这是因为它给习惯从左至右书写的用户造成了阅读障碍。

会议通知

精伦电子第二届董事会第四次决议公告暨召开股东大会的通知
《上海证券报》第30版 2007-2-27

会议召集通知预订

股东姓名 张学阳 *
密 码 *
股东身份证号码
股东持股数
受托人姓名
受托人身份证号码
发送邮件通知 是 否
备 注

提 交 取 消

不管是左对齐还是右对齐，你并不能说谁更好谁更差。这方面一直就没有一套可以区分高下的通用标准，有可能将来也不会有。

我对此的准则是：如果用户不需要对控件标记仔细辨别（例如那些他们熟悉的内容：姓名、密码、住址），用**右对齐**比较合适，因为他们更多注意的是**右边**的控件；而如果你认为他们的注意力主要放在**左边**（例如那些需要看清楚的内容：股东持股数和受托人姓名），强调一下**左对齐**会比较好，因为这样便于他们检索需要输入的内容。

不要只注意一边

不光是左边控件标记需要注意对齐，控件本身也要注意对齐。下面这个例子就很难看：

填写用户信息	
用户名*:	
密码*:	
密码确认*:	
用户姓名*:	
E-Mail*:	
联系电话:	
用户描述:	

大多数时候设计人员只是根据某个控件本身的字符长度来显示控件的外观长度。而每个控件的字符数设置又是很随意的。这就导致控件长短不一。用户对这种“不负责任”的显示方式并不欣赏。

难道下面这样不是更好看些吗？



如果有时候左右都不能对齐

不管你设置的是左对齐还是右对齐，有时候你还是会遇到一些特殊情况，例如某个控件的标记非常长。如果在这种时候仍然拘泥于对齐方式的话，就很可能会造成空间的浪费。



如果你觉得把这句话分成两行不合适，也不能把字数简化，那么最简单的办法就是忽略这个标记的对齐方式，其他地方能对齐则对齐。就像下图所显示的这样：



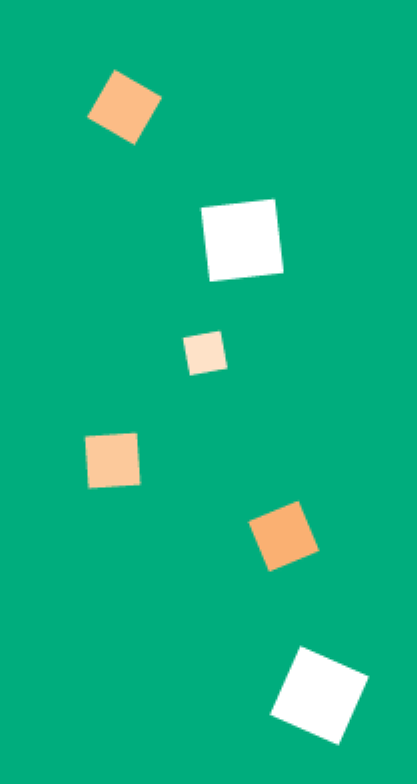
上下也要对齐

除了左右对齐之外，你也需要注意上下水平的对齐方式。这一点很多设计师都没有加以足够的重视。

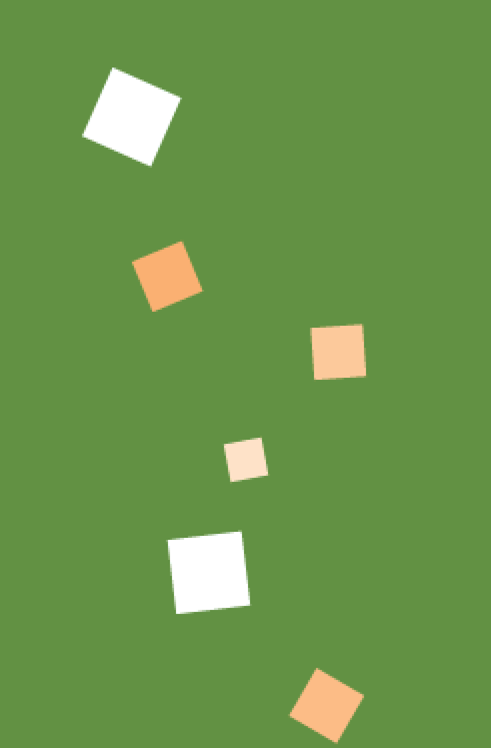
例如在 Gmail 的邮箱设置页面中，面板第二行的单选按钮、文字、文本框和下拉框全部不在一条水平线上。这给人一种不平衡的感觉。下面的另一个下拉框就好多了，尽管它也没有与文字完全水平对齐：



我相信这个问题并不难解决。一直以“Focus on the user”自称的 Google 居然也会出现这种蹩脚的失误，实在是令人有些难以置信。



第八章
测试可用性



至尊宝：“出来吧！葡萄”
菩提：“我不是想监视你，我只不过是想研究一下人与人之间一些微妙的感情。”

——《大话西游》台词

尽管我们已经讨论了许多网页和 Web-based 产品可用性方面的问题，举了很多例子，但是这些还不够。

也许我们可以用一整本书来穷举所有可用性方面的错误，提出针对这些错误的改正方法，就好像一部由“错误集合”所组成的某种“武林秘籍”，但这些也仍然不够。

这是因为就好比我们无法把人类的特性描述得面面俱到一样，每个人都是彼此不同的。可用性也是如此，我们永远无法通过某种“规则”或者“秘籍”来避免所有的错误。不同的用户在不同的时间、不同的动机和意图、不同的情绪状态下表现出来的特性也是不一样的，没有人能够通晓一切。

那么应该怎么办呢？如何确保我们的设计能够让用户满意？你得经过怎样的证实才能有把握自己设计出来的产品能够经受得住用户和市场的考验？

答案只有一个，测试。

8.1 为什么要测试？

如果现在的设计中有个地方存在着两种(或者更多)解决方案。你该如何选择？哪种方案最好？

不论设计者自己怎样费尽心思考虑这个问题，也不论团队（假使是团队）的所有成员如何热闹地聚在一起讨论，他们很有可能到最后还是什么都解决不了。哪怕这个过程得出了某个结果，你也无法凭此就相信它是正确的，或者是最好的解决方案。

这种思考或者讨论，其实只是在浪费时间。

8.1.1 美国到底民主还是不民主?

这种浪费时间的讨论我相信你肯定有过类似的体验。例如下面这种情况。



这种争论有点类似关于“全球变暖到底是人为造成的还是地球自身的变化规律？”、“美国到底民主还是不民主？”或者“苹果到底是削了皮吃好还是连皮一块吃好？”的讨论。这些讨论往往没有结果，每个人都有自己的看法和依据，但同时又没办法说服对方。最后每个人仍然保留着自己原来的看法。

而与那些争论不同的是，在平时的争论中彼此无法统一认识也就算了，不会有什么影响；但是在设计工作中，我们必须得出一个结论来，所以总得有人服输。这就很可能产生紧张气氛，破坏团队成员间的关系，导致无法做出关键的决定。

这似乎是无法避免的事情。为什么会出现这种情况？

我喜欢的，别人也应该喜欢

设计师都有双重性，一个是他作为设计师的一面，通过对用户的调研和沟通来设计适合用户使用的产品；另一面则是作为使用者一面，通常他们都把自己假想为用户（也可能他们本身就是真实的用户），根据自己的意愿和喜好进行设计。

每个人都有自己的喜好和憎恶。你可能喜欢网页中的下拉列表，因为它们浅显易懂而且不占地方；你也可能讨厌下拉列表，因为有它们出现就意味着可能要在长长的列表中找寻目标。我喜欢伸缩式的动态网站导航，因为很有趣味性；我也可能不喜欢它们，因为无法对网站的结构一目了然。

这样的结果是希望自己设计的产品能够满足自己的喜恶。我不喜欢下拉列表，那么我就在页面里面找其他方式来代替下拉列表。

这是一种天性——“我是正常人，而且广大用户也是正常人，那么他们就会喜欢我所喜欢的。”有时候我们也明白在有些问题上某些人可能和我们并不一致，但是那种情况一般并不多见，而且最主要的可能是因为他们并不像我们这样明智——不明智的人总是存在的，我们团队里面就有几个——那些跟我意见不统一的家伙。

在这种思路下，设计师总是能够“自我安慰”，然后在设计中加入过多的个人意愿。这种结果往往就是一大堆人看着设计稿，每个人都有自己的主张，尽管他们的目的都是为了做得更好。

角色的不同导致看法不同

另一方面，这种对立的看法也来自于开发团队中的各个角色职位的不同。

设计师们通常认为，大多数人都喜欢漂亮的、有趣的界面，因为设计师自己就喜欢这样的界面。他们总是找寻视觉上令人感觉新奇的网站或者界面，仔细地欣赏和品味，有时候甚至把它们截屏并保存下来。

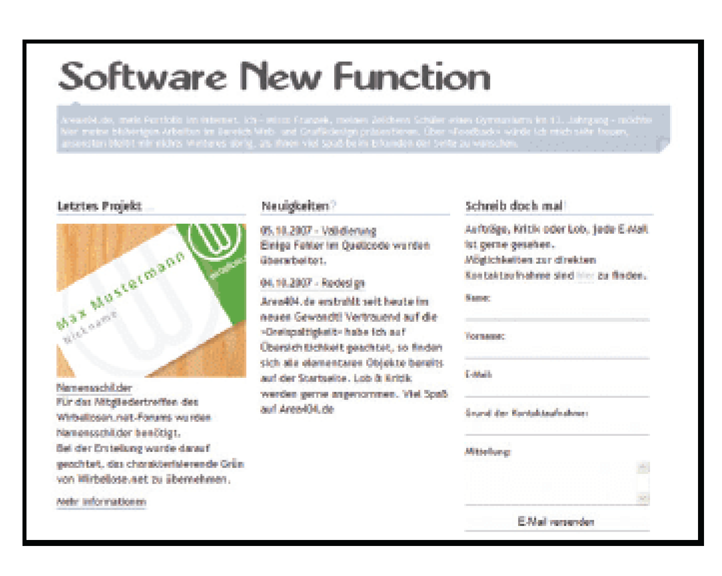
而另一方面，技术开发人员则认为人们都喜欢页面简洁，同时功能又多又新颖的产品。因为他们觉得这样才是最好的，体现了技术力，科学而严谨。

这样一来，设计师们想做又美观又酷的有趣界面，而技术人员则想做功能完备的新潮产品。哪个部分才是开发中最应该优先考虑的呢？这就是大多数美术设计和程序设计人员之间总是无法互相认同的原因——他们经常会因为各自的喜好而对产品特点产生对立的看法，有时候甚至相互发生冲突。

而一旦有市场人员，或者老总出现的时候，设计师和技术人员会突然发现他们站在了一条战壕里，共同面对的则是游戏里面的终极 BOSS——管理者和市场文化。这些家伙关注的是投资者、浏览量、广告收入和战略合作伙伴，他们并不为产品本身的美观或功能操心。至于盈利等等这些责任就落在了设计师和程序员这样的人身上，让他们觉得自己卖命的所得似乎全部落入了资本家的口袋。



设计师最喜欢这样做



开发人员觉得这样最理想



市场人员感觉这样最好

有时候从小事里面能够看到更深层次（同时也更难解决）的问题——这就是艺术、科学和商业长久以来一直就存在的矛盾。

8.1.2　这不是简单的是非题

在我们彼此争论的时候，总是会说“我觉得这样好，因为大多数用户都喜欢这样”或者“大部分人不喜欢这玩意，所以我们把它去掉吧”。然后对方可能马上反驳：“谁说的，我认识的人都不习惯这样做，他们都喜欢那样。”

然后讨论开始转化为到底大部分用户都喜欢什么或者不喜欢什么。似乎只要找出人们的喜好，就能做出最好的设计。没错，在大的方面或者趋势上面我们能够得出一些结论，例如导航栏一般都放在页面的最上方，例如大家都开始喜欢把个人网页做成 blog 的形式，或者现在界面的美术设计都偏向晶莹透亮的材质，就好像 Vista 或者 Mac OS。这就好像在唐朝的时候大家都喜欢胖女人，但现在更欣赏苗条一样。

但是对于大多数可用性细节方面的问题，并没有一个“主流”或“非主流”的比较。这种比较看上去把问题简单化了：要么就是好，要么就是不好。超链接的下划线要么很好（它区别于其他普通文本），要么不好（太多的下划线让人眼花）。弹出窗口要么很好（它吸引了我的注意力），要么不好（我还得花费时间去关闭它）。应该把数据结构表现为树状列表，或者不应该在产品里出现树状列表。总之不是黑就是白，不是对就是错。

但是，这里其实没有简单的是非题。我们不能判断“大部分用户”是喜欢这还是喜欢那，因为没有人能证明喜欢A种方式的用户总数比喜欢B种方式的用户总数更多。每个用户都是独一无二的，他们不会就这些问题为我们形成统一的意见。

例如，很多人习惯网站中的弹出窗口，但同时又有相当数量的人声称他们非常讨厌浏览网页时弹出窗口。但实际上，这些人讨厌的是那些使用不当的弹出窗口：莫名其妙地突然蹦出来，满是耀花了眼的广告，即使是真正意义上的对话框，却又在同时禁止了其他的所有操作，而且用语冷漠令人生厌。

如果你仔细观察这些用户，并且正确地询问他们，就很可能发现其实这些人并不反感那些正确使用的弹出窗口：适当的提示或告警，有意思的便捷入口，以及友好的用语。有的弹出窗口他们甚至很喜欢，有时候他们都意识不到网站弹出了某个窗口。

其他地方也是一样。有些人说他们讨厌 Flash 动画，甚至专门建立了讨伐 Flash 的网站和论坛，但他们反感的是那些没有带来任何价值，却又耗费他们大量时间去下载的 Flash，或者闪得让人头疼的 Flash。对于那些经过专业设计、尺寸精巧的 Flash，或者有着良好体验的 Flash，他们其实很喜欢。这些事情没有绝对的爱与憎。

那么如何解决关于 A 好还是 B 好的争论呢？

解答的关键不在于一定要弄清楚“是不是大部分人喜欢 ×××”，这种问题本身就是不正确的。你应该问“在这个页面中，当用户执行这类的任务时，××× 放在这里会不会让用户感觉良好？他们会不会感觉更方便？”

不是 ××× 对或者不对，而是放在这里合不合适。

而要想回答这个问题，解决方式也只有一个，那就是可用性测试。我们根据用户的需求来提供几套设计方案，然后把设计方案制作成原型，再把它们展示给用户，看看他们的看法是怎样的。如果大多数人都觉得 A 好，那么就是 A 了。

这是唯一的方式。

我们可以看到可用性测试的几个好处：

- 能够在产品正式上线之前（而不是之后）发现并修复错误。
- 设计团队能够专注于真实问题，而不是自己假设的问题。
- 设计师和技术人员能够专注于设计和开发，而不是相互争论。
- 能够缩短产品的开发时间。
- ……

另外可用性测试还有一点好处，就是能够让开发人员相信：不论多么有信心的设计，经过用户测试就肯定能发现还有一些问题需要纠正。很多设计师和技术人员从心里抵触修改的建议，其中一部分原因是出于修改需要花费更多的时间和精力，另一部分是因为一种看法，就是害怕会把出现问题的责任归咎到他们头上。

但实际上，你要记住下面这句话：

> 可用性测试是一种基本的开发工具，
> 它不是评价开发人员的手段。

在可用性测试中发现问题是正常的，如果没有任何问题倒反而不正常。很多在一开始抵制可用性测试的开发者在经历了几次测试之后都变成了虔诚的信徒，他们积极寻找机会测试他们的设计，从而获得改进的反馈。

用户界面设计不是艺术，需要神秘的创造天赋，在灵感激发瞬间的火花中看到一个完美的结果。这种设计是一种基于科学、实践和行业标准的工程学科，它要求得到反馈信息，而这些反馈必须通过对用户的观察、合作以及测试来得到。

可用性测试可以说是一种很基本、很常见的开发工具，这种工具存在的目的就是为了找到并解决更多关于可用性方面的问题，从而确保尽可能多的用户能更容易地接受我们的产品。

8.2 几个要注意的事实

关于可用性测试，我认为有一些需要注意的事实。

8.2.1 一些需要记住的事实

在这里我列举了一些关于可用性测试方面的正确认识，理解它们是非常重要的。

可用性测试不是软件测试

有些人很容易就把可用性测试和软件测试弄混。这是因为它们都是测试，针对的对象都是有一定完成度的产品，同时也都需要专门的测试人员，会提交完整的报告。它们都是一种工具或手段，目的是为了让产品的错误更少、表现得更完美。

也许有些人会说，我们一直有专门的软件测试团队，他们工作得很好，所以我们不需要再搞什么可用性测试——反正看上去它们也差不多。

但毫无疑问，可用性测试不是软件测试。

- 在**软件测试**中，一定数量的测试人员不停地操作软件完成各种任务。重点在于程序、技术和逻辑，目的是为了检测软件是否能够实现所有预期的功能，以及寻找 BUG。测试人员需要自己记录测试结果，然后提交到程序人员手中。

- 而**可用性测试**中，测试人员这次是一个普通的用户（也许会是专家，但他们也会让自己“扮演”用户的角色），并有许多（可能是隐匿的）旁观者。用户操作产品完成各种任务，他们要说出在这个过程里面是否感觉良好。有专人负责记录，可能还需要录音和录像。

软件测试是针对于产品本身质量和功能的评估。如果出现了问题，就表示产品根本不过关。没有通过软件测试的产品完全不能发布——不然那简直就是笑话。

但是可用性测试更注重于用户使用产品时的感受。以用户的角度，让用户自己来对软件进行评估，侧重点在于可用性和良好的用户体验。也许不经过可用性测试的产品也能够推放到市场，但是如果使用起来不方便，用户的口碑也不会好到哪里去。

有时候，一个网站可能不需要软件测试，例如纯静态的网页，不会存在什么程序方面的问题，但是它却需要可用性测试来判断浏览者是否在网站中感觉舒适和轻松。这是一种偏向于感觉意义上的评估。

可用性测试不是要取代质量测试，只是在补充和增强它们。

哪怕只测试一个用户，也比不做测试要好

只要进行可用性测试，哪怕就一次，你也会得到很多收获。

想想你为之工作了好几周甚至好几个月的产品。你对它了解得太多了，哪怕那些不正常的地方你也会觉得再正常不过。这可以说是另一种意义上的“习惯”思维模式。但是用户和你不同，他们没有接触过这个产品，他们能以一种正常的心态，以客观的眼光来体验产品。

这就好像你家乡的风景区或者动物园。也许你从小到大去过不下 50 次了，你对它再熟悉不过，

每一个景点、每一种动物。但是如果你邀请外地的朋友来参观，你必然会发现一些平时不会注意到的情况，或者他们会提出一些你没有发现过的好玩的事情——哪怕他们对于历史或动物完全是白痴。在这个过程中，你会意识到很多你认为理所当然的东西，对别人来说却并非如此。

早点测试一个用户，好过最后测试 100 个用户

很多人都有一种错误的印象，他们觉得可用性测试应该在网站或软件产品几乎就要发行面市的时候进行。这有两方面的原因：

- **他们觉得用户测试很复杂**

  他们觉得需要大量的准备工作、招募测试用户，还需要各种精细的测试工具和设备——既然这样，不如在最后来一次大的，一个锅摊好所有的鸡蛋，一了百了。

- **他们觉得用户测试也很容易**

  他们认为测试只要能顺利地开始和进行，就可以很容易进行修改。在最后来一次大的，把问题都挖掘出来，只要发现问题就去改正，没什么大不了的。

但这都是错误的理解。第一，用户测试没那么复杂。关键是，如果你把它们想得很复杂，或者设计得很复杂，你就不愿意尽早进行测试，也不会进行充分的测试。如果没有充分的测试，就无法从中获得最多的收益。早点开始测试，哪怕很简单，你也能尽早地发现问题，这种时候还有机会用得上这些测试结果进行修改。它非常有价值。

第二，用户测试也没那么容易。一个锅摊好所有的鸡蛋？那就意味着你将一次性得到海量的测试数据。研究这些数据可没有你想象的那么简单。



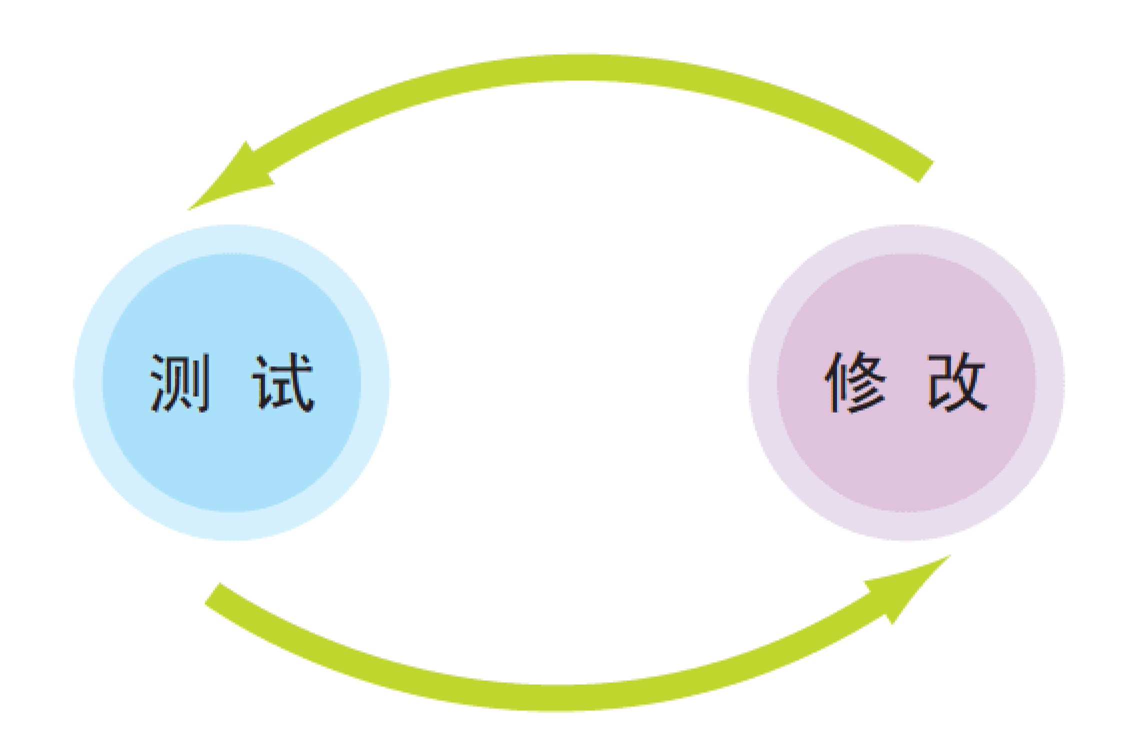
我们每个产品都进行一次大的用户测试。
现在研究人员正在对付那些测试后的数据。

第三，事实上最后的测试对于待发布产品的影响力几乎为零。在这个时候所有人都非常渴望发布产品。有时候一些经理在这种时候雇佣专家来评估和测试，但是当他们拿到了评估报告和改进建议后，他们说："这里面大多数的问题我们都没有时间改进了，非常感谢你的报告，酬劳当然会照付。再见！"

而且对于网站来说，一旦它已经投入使用，再要想改变它就不是那么容易了。有些用户拒绝做出任何变化，即使很小的变更也会给他们带来深远的影响。另外这也是件伤筋动骨的事情，想想中国电信更换 logo 时花费了多少人民币？

"迭代"的测试和频繁的测试

可用性测试不是做一次就可以了，而是要遵循测试 -> 修改 -> 再测试的循环过程。



用户在测试过程中发现了某些问题，我们必须要回到设计方案阶段对这些问题进行修改，并再次制作修正后的版本。此时需要**再次**展示给用户，让他们进行再次的测试和评估。而这次评估的目的有两个：一是检验这次的修改是否合乎要求；二是看看是不是还会发现其他新的问题。如果第一次发现的问题已经修改并解决了，用户就会把注意力放在其他方面。

测试从没有"太早"这一说，你需要在每个阶段对每个目标都进行测试。测试就是检验，如果每一次检验都得到比较好的结果，那么就能确信自己始终走在一条正确的道路上面。

- **正式开发之前**

  提供一些初级原型,以此来确定概念设计的大致方向。这个时候由于不是一个精雕细琢的设计，用户不会被那些细节所吸引，从而可以把注意力集中在那些最关键的部分。

- **开发过程中**

  这个阶段开始制作高级原型并且不断地根据用户反馈对其进行修正。用户们更愿意评论一些看起来还没有完成的东西，因为他们知道你还没有投入太多，所以愿意大胆地提出一些意见——你还有机会进行修改。

- **开发之后**

此时再进行一些用户测试为发布之前清除较小的可用性问题提供指导。有时候在这个最后的阶段也会暴露出一些“终结者”的角色，它们如此重要，以至于必须推迟面市的时间进行修改，也就是“跳票”。

有些人可能会认为可用性测试很耽误时间——开发进度本来就很紧张，哪里还有时间能抽出来做测试？但其实只要测试做得好，你便能节省时间。因为它能够避免设计过程中无休止的争论。而且最关键的是，这样能够确保你前进的方向始终是正确的，而不是越走越偏。它避免你在最后去返工。

测试后马上回顾测试结果

在每轮测试之后，你应该尽快与观察者和被观察者共同检查所记录的笔记和其他数据，同时研究细节，并且找出那些还有些含糊的地方。

这是因为人的记忆能力有限，也许很多当时很清楚的认识在第二天就模糊不清了。而且，如果只测试而不马上回顾，记录就会积累得越来越多，很可能到最后根本没有时间和精力去在那么多数据中找到你想要的。

因此，一旦进行了测试，开发团队就应该马上进行总结。这样的总结工作包括两件事 ：

- **把问题进行分类**

和测试用户一起回顾看到的问题。讨论问题出现的原因，并决定哪些问题需要修正。

- **解决问题**

共同讨论修正这些问题的方法。

用户提出的看法和他们碰到的问题其实非常有说服力。在最初彼此争论互不相让的开发团队成员会发现，他们在用户身上能够统一认识。同时，测试是一个循环迭代的过程，因此不必对完美的解决方案达成一致，你只需要确定下一步做什么就可以了。

8.2.2 一些需要避免的认识

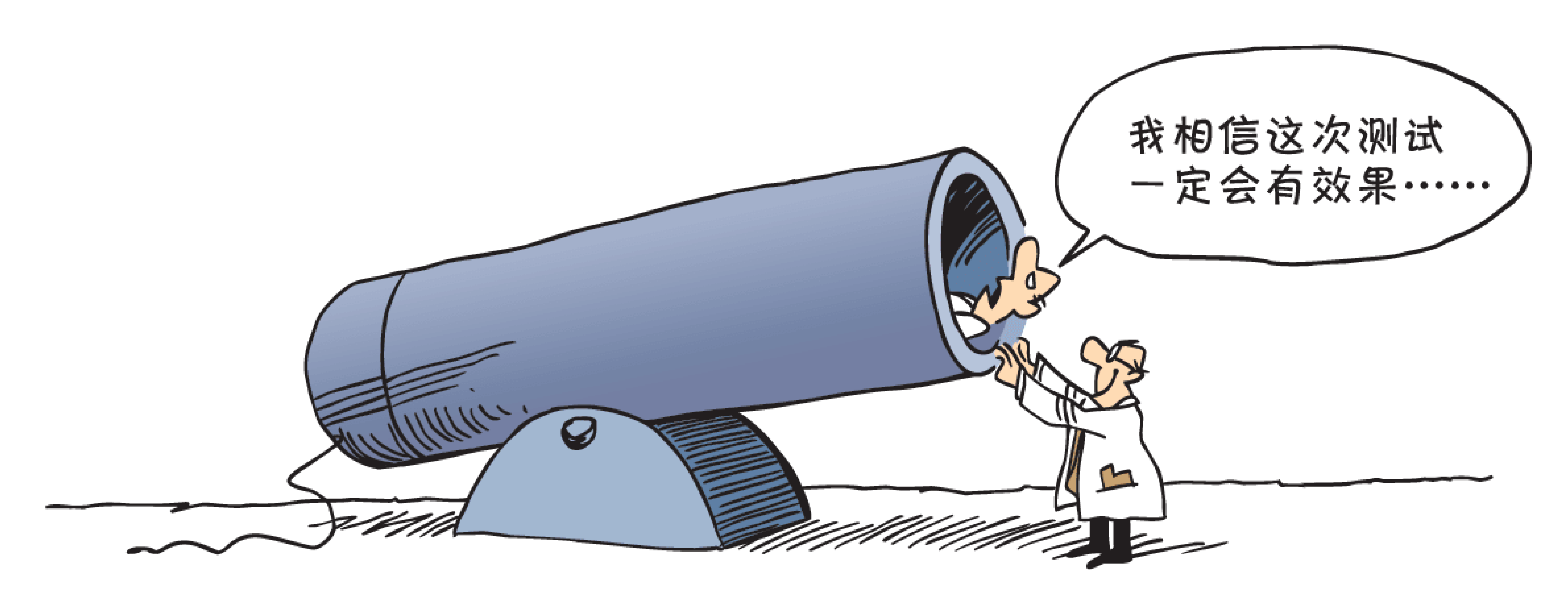
此外我们还需要知道一些惯常的误解。它们从另一个方面说明了一些问题，有助于你更全面地认识可用性测试。

不要认为可用性测试很难

可能你会有这样的疑问 ：“我没有专业的知识，有能力完成这种专业的测试吗？”

其实很少人知道，可用性测试容易得令人难以置信。它只不过就是与参与测试的用户进行沟通，循循善诱就行了。也许确实有些人会比另外一些人做得更好，但是我从来没见过一次可用性测试没有产生有用结果的，不管它进行的有多么糟糕。

测试总是会有效果，哪怕是对错误的用户做的最糟糕的测试，也会让你看到一些改善可用性的重要方面。



不要认为可用性测试总是非常昂贵

很多人认为可用性测试花费昂贵。在堆满精致设备的专用实验室里，观察者坐在半镀银的大镜子后面做着笔记，花钱专门请来的用户在测试间里面对着至少两台摄像机，使用产品的第 N 个版本来完成某个任务。

但这是十几二十年前的可用性测试。那时候要进行这种科学研究确实至少需要花费 6 万以上人民币。在今天，也可能还有些非常严谨的实验仍然是这样来做的。但是更多的测试已经花费更少，也更为快捷。最简单的可用性试验设备就是一台电脑和一个安静的房间。

我们不需要太专业的设备。现在的小型家用数码设备完全可以取代以前笨重的仪器。至于聘请测试用户的费用，你也可以通过找一些朋友代劳来节省，而酬劳可以是一块巧克力或者一件 T 恤衫。别误会，我没有不尊重测试用户的意思。如果是正常意义上的参与者，你应当付给他们酬金（通常是数百元人民币），以补偿他们在测试以及路途往返中所花费的时间和精力。

另一方面，跳过测试并不能真正节省费用。也许取消了可用性测试确实能够节省一部分的钱，但是把公司当作一个整体来看就不是这样了。不做测试省下的花销将会用在日后的客户支持上，也许你得花几倍甚至更多的钱。

在开发阶段取消测试，就意味着把产品拿到市场上去做测试。那个时候哪怕很微小的失误也

可能会让你所有的开发投入都打了水漂。

关于如何在可用性测试中省钱，我们在后面会详细地讨论。



打了水漂的钱可是收不回来的。

不要认为可用性测试是表层问题

相当一部分关于用户界面的误解，集中在认为“用户界面就是美术设计和软件表层的方面”。这些人认为可用性测试只能发现一些表层的问题，例如对话框的布局、按钮的位置和颜色等等。这些问题都很好解决。

但实际上，可用性测试是一个深层次的问题，可能会带来产品的重新设计或者结构的整体改变。

受到之前那种错误思想的束缚，有很多经理们在看到真正的可用性评估报告后，会感觉奇怪和头疼。他们会感到好像触摸到什么很深、很黑暗的东西，隐约感到将要付出相当的代价才能做到真正的改进。而这让他们很难去接受。这能够解释为什么很多经理在进行了可用性测试之后，却并不打算解决发现的主要问题。

还有一些设计师和技术人员拒绝做某些用户界面的改进工作，托辞是没有时间，但同时他们却在改进图片和铃声，因为他们觉得这个更“酷”更好玩。然而在产品本身可用性没有得到提高的前提下，任何有关“皮毛”的修改都没有多大意义。

不要认为测试是要去“证明”什么

有些设计师和技术人员，包括他们的经理，认为做可用性测试是要“证明”用户界面的完美。这些人并不了解可用性测试。他们对自己设计并开发的产品非常自信，根本就不认为可用性测试能够找出有关产品的任何问题。

在这种心态下，即使有问题出现，他们也不会花哪怕一丁点时间去解决。

这是因为，他们觉得如果测试的结果与预期的相违背，那么肯定就是测试本身出现了问题：要么就是测试没有设计好，要么就是参与测试的那些用户太愚蠢。

其实这背后还有更深层次的原因，就是他们害怕出现问题，认为出现了问题就表示自己没有做好，因此他们拒绝任何问题。但这同样也是错误的。我们在之前就已经谈到，可用性测试与评价开发人员的水平毫无干系。

不要仅仅测试，却不纠正发现的问题

这看上去好像是一句废话——仅仅测试可用性是不够的，我们当然要对测试中暴露的问题进

行修改和纠正，否则测试有什么用？

但是这却是个很常见的情况。我见过很多团队在开发周期预算的时候，根本就没有考虑改进可用性所要花费的时间。而即使他们考虑了这个问题，也顶多估计为大概一到两个工作日。

造成这种情况可能有两个方面的原因 ：

- **不得已才做测试**

  有些开发经理安排可用性测试，仅仅只是因为它是公司开发流程中的一个强制执行的环节。他想做的只是保证这个步骤已经执行了，但并不关心评估的结果。

- **他们认为改进很简单**

  这有点类似把可用性问题当成是表面问题。他们觉得整个产品的程序和架构都已经成熟了，要改的只有一些小的局部，那不需要多大功夫。



某个医院软件产品的可用性测试

这种不负责任的决策所导致的问题就是，尽管会发现很多问题，但是你在匆忙中根本无法根据测试结果和反馈意见做出有针对性的良好改进，导致可用性测试最后其实成了一纸空谈。

8.3 测试的前期准备

没有前期的充分准备，谈不上开始测试。这就好像你不去买牛肉就做不出牛扒。

如果你现在想看看自己做牛扒的水平怎么样，你会需要些什么？首先你需要有个厨房（哪怕很简陋，没有抽油烟机），然后你得有灶和煎锅，而且你需要几个愿意品尝你手艺的人。当然，很重要的还有厨师自己——你要站在一旁仔细揣摩食客们对牛扒的反响到底如何。

我们分别来看看这些准备工作如何安排。

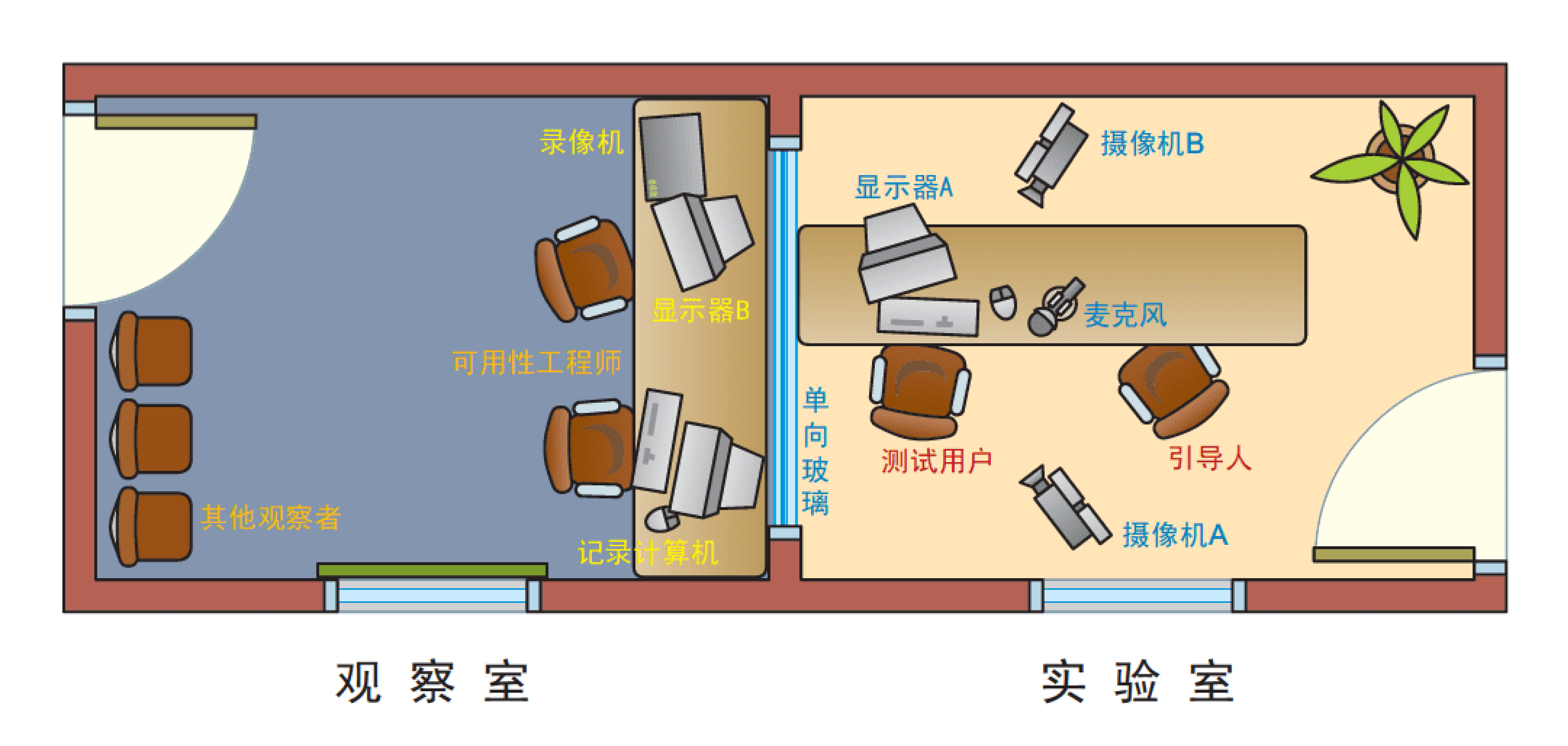
8.3.1 测试的地点和设备

可用性测试需要一个专门的空间，以及必要的设备。我之前提到能够让测试变得更加省钱和快捷，但首先还是来看看最正规的可用性测试需要些什么——没准你的公司有能力而且正好需要这些玩意。

最正式和最精良的装备

一个完整的可用性实验室由两个房间组成：实验室和观察室。这两个房间之间是一面单向透光的镜子，而且墙壁是隔音材料。一般来说观察室墙壁通常是深色（确保不会透到镜子那边去）。我们在国外大片中经常能看到警局里这样的装置（很有个性的罪犯会故意对着镜子做鬼脸）。

参加测试的用户在实验室内的计算机上使用被测试的产品、软件或网站，观察者们坐在观察室里进行观察、记录，有时候也会讨论。隔音的镜子和墙保证用户不会被干扰。必要的时候他可以通过麦克风和观察者交谈。也有些时候会有一个测试引导人与用户一起坐在实验室里，由引导人根据事先的安排来主导测试的进行。



实验室中的计算机连接着两台显示器。显示器 A 放在用户面前，显示器 B 则放在观察室。A 和 B 显示的内容是同步的。测试用户进行操作和交互的全部细节都同时显示在这两台显示器上。

此外，在实验室会有一部可以遥控的摄像机 A，用于记录用户桌面上的键盘和鼠标动作。你通常还会看到第二部摄像机 B，用来对准用户的脸，观测他的面部表情。也许还会有更多摄像机。

实验室里面的麦克风和摄像机所接收到的信号被传送到观察室里。有时候用户计算机显示器上的信号，会连同声音和摄像数据一块被转录到录像带上进行保存。这需要通过一个扫描转换器处理。

此外，可能在观察室还有一台计算机，用于方便观察者们输入观测记录，例如用户的评论、动作、

表情和相对应的时间。

大概就是这么多。虽说现在这种实验室并不是测试的必要条件，但“工欲善其事，必先利其器”，一个装备完善的可用性实验室可以帮助你更多地进行可用性测试——既然提供了这样一个一流的地方，想怎么做都可以。而且，一个设备精良、体现出高科技和财力水平的实验室，将非常有利于提升公司的形象。想想看，我们公司是多么重视产品可用性——而且，它比较含蓄地证明——我们公司是多么有钱。

这种装备精良的可用性实验室造价大概需要花费 6 到 15 万元人民币。

也许有点贵？而且它看上去更像某种庞然大物，高科技的尖端产品，至少需要两到三人的协作、周密的安排，以及复杂的操作和控制。但是总有别的办法。人们总有新的办法来解决问题。

你也可以优化和精简

我们可以直接举出最“极端”的简化方案。你可以根据实际情况（金钱和人员配置）来加以权衡，看看在哪些地方可以简化，哪些地方又可以稍稍偏向于标准一些。这个天平掌握在你自己手里。

- **关于实验室**

两个单独隔开的房间是有必要的，如果在众目睽睽之下做某件事情，而且还是自己很不熟悉的事情（如果他很熟悉，那就表明你该换一个人），任何人都会非常紧张。

但是没必要特意去开辟一个中间有单向玻璃的专用实验室。任何一间办公室或者会议室都能开始测试，引导人和用户坐在一起，而其他观察员坐在附近另一间办公室里。他们在那里怎么讨论都没关系。

有时候甚至可以让引导人兼职观察员的角色，尤其是对于网站这种并不复杂的测试。这样第二间办公室也不需要了。

一个安静的会议室或者办公室就足够了。所以这项花费为 0 元。



● **关于两台显示器**

实验室和观察室需要靠近的唯一原因在于，测试用户使用的计算机需要两台显示器，而显示器的连接线一般不会很长。

但双显示器并不是必需的。观察室里的显示器无非就是要同步显示用户的操作屏幕，你通过远程桌面连接完全能达到这一目的。或者也可以利用一些软件来实现屏幕共享，例如微软的 NetMeeting，也许还有其他一些更好的软件。

双显示器并不花什么钱，但是你完全可以避免这其中产生的麻烦。这一项的花费同样也是 0 元。

● **关于摄像机和麦克风**

摄像机用于录制用户与硬件的交互操作，例如键盘和鼠标。此外还有他的面部表情。如果有摄像机，它就能录音，那么麦克风也不需要了。

如果你没有这么多预算，你完全可以找人借一部数码 DC。也许你自己家里就有。

我认为摄像机并不十分必要。用户面对着摄像机时可能感觉不太自在，而且网站或者 Web-based 软件不是什么新奇特殊的交互设备，一般来说没有多大必要记录他们与计算机的交互过程。如果有什么特殊的情况发生（例如快捷键很难使用），引导人很轻易就能发现。另一方面，录像会让整个测试变得更复杂。

如果你坚持希望同时看到用户的表情，也有更节省的办法——买一个小摄像头。把它放在显示器上对准用户，观察员哪怕用 QQ 这样的软件也能同步监测。

如果你的录像设备不支持录音，那么就可能需要录音设备。但没必要使用专门的麦克风，这不是在录制唱片。现在市场上几乎所有的便携式笔记本电脑都有拾音孔，如果你觉得这还不够，买一个耳麦也行。或者把你那个带录音功能的 MP3 拿来暂用一下。

那么，你需要买一个摄像头，一个耳麦。费用不会超过 200 元（好吧，你希望要 SONY 或者 SHARP 的原装货，800 元也足够了）。

● **关于录像机**

录像机用于合成测试过程中显示器、摄像机和麦克风捕捉的信号，并且把它们转录成录像带以供存档。显示器的信号数据需要通过一个扫描转换器处理后才能被记录。

关于这一点没什么好说的。如果你的计算机有足够大的硬盘，完全可以把那些声音和视频数据直接存储在计算机里。根本用不着录像机。

看看，除了必备的电脑之外，你总共需要的只有一个摄像头和一个带麦克风的耳机。有的笔

记本电脑甚至自带摄像头和麦克风。

哪怕你穷得揭不开锅，也有适合你的可用性测试。找一个其他部门的同事，让他坐在电脑前使用产品。你同样也能发现问题。用不着摄像头和耳麦，你需要的只是一支铅笔和笔记本。

8.3.2　测试用户的招募

牛扒现在端到了餐桌上，你需要几个人来对它做出评价。只去高档餐厅吃法国牛扒的人可能不太合适，他们的口味可能已经被惯坏了；那些从没吃过牛扒的人也不太合适，他们的意见没有可对比性。

对于可用性测试来说，正确选择参与测试的用户同样是一个问题。你不能随便拉几个人来就完事。

测试用户的选择

对于可用性测试，我们必须选择有代表性的用户。他们应该能代表产品现有的和潜在的目标用户群体。你可以从参加者的专业知识、计算机经验和对产品的熟悉程度来进行选择。

- **用户是否了解必要的专业知识？**

  有些产品是为专业人士设计的，例如一些网络财会软件，或者保险公司的专用在线系统。找一些对这些产品相关知识一窍不通的人来做测试几乎是没有意义的。而另外一些产品则没有专业知识要求，例如门户网站。

- **用户的计算机经验如何？**

  如果参加测试者的计算机知识过于贫乏，可用性测试就会变成用户学习使用计算机的测试。这应该由那些硬件厂商来做。另一方面，普通用户和程序员对软件的理解会很不一样，他们彼此的测试结果也会不一样。

- **用户对产品的熟悉吗？**

  没有使用过或是很少使用被测试的产品的人属于新用户。多次使用而且对产品非常熟悉的人属于熟练用户。而大多数用户则介于这两者之间。

  对新用户进行测试，你会发现产品的易学性和易记性方面的问题；对熟练或半熟练用户的测试则能发现有关产品可用性、效率和用户体验方面的问题。这都很重要。

此外还有其他一些需要考虑的因素，例如参加者的性别和年龄等等。你需要针对目标用户群体进行分析，确保参加者能相对平衡地代表所有目标用户。

另一方面，由于合适的测试对象数量少、工作繁忙或者索价太高等原因，你有时候会发现很难请到那些最有代表性的用户参加测试。例如某个专业领域的产品需要咨询那些专家的意见，但他们总是很忙，或者不屑于参加测试。这确实是个很令人头疼的问题。



—您的意思是参加测试需要5位数的酬金？
—现在博士参加测试都这个价。

这并不意味着可用性测试就无法进行。我们可以考虑采用一些办法来克服这些困难。

- **有些问题并不需要行业专家来测试**

并不是所有的产品都需要专家来测试。网站之类的产品就不用说了，即使是一个农林反病虫害控制系统，不是农药施放管理员也照样可以测试里面的电子地图定位模块。或者一个机场调度的管理系统，你也不需要一个真正的调度员来测试里面的表格和控件是否能正常工作。

- **即使需要专家，也可以降低一些要求**

你可以请一些近似于专家的人来代替专家。这些人索价更低，也不会总是忙得抽不开身。例如你可以请计算机系的学生来代替专业程序员，实习生代替正式职员，助教代替教授，经理代替总裁。这些近似于专家的人一样有很高的专业性。

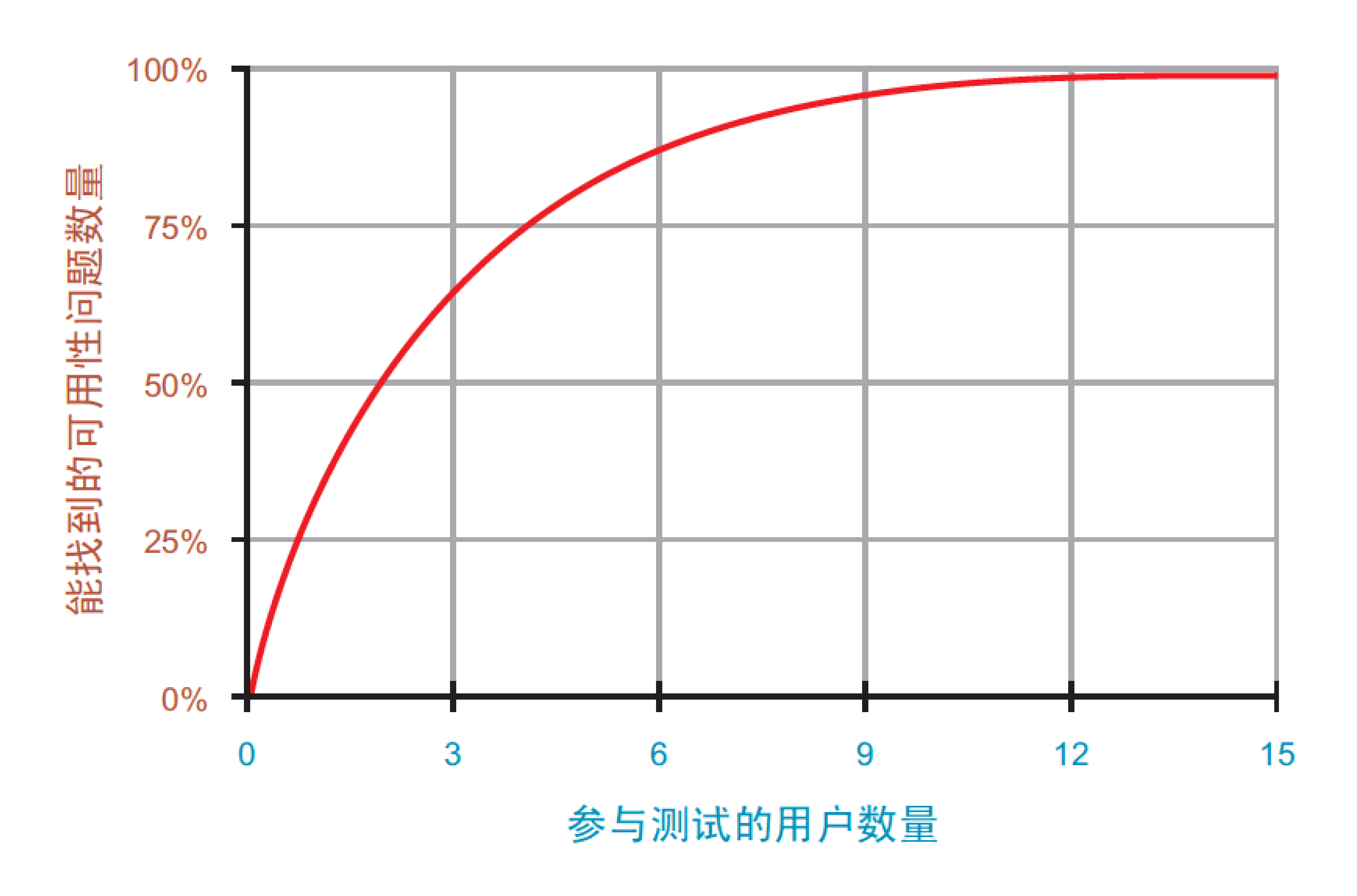
- **非要专家不可的时候**

在这种山穷水尽的时候你只能去邀请他们。坦白说，这可能意味着需要更多的付出，例如咨询费用。但是，如果你能让那些专家觉得这个产品对他们来说有很重要的意义，会对他们的工作或研究产生好的影响，他们也许会很愿意参与测试。

究竟每次应该测试多少用户？

谁能说明要测试多少个用户才算是一次充分的测试？ 20 个？ 50 个？

别想得那么恐怖。看看 Jakob Nielson 博士的这张经典的图表。它标明了参与测试的用户与发现可用性问题之间的数量关系。



忽略其他的数据吧，看看我们的结论：

> 有5名用户参与的测试，将能发现85%的可用性问题。

事实也确实是这样。一次测试 4 到 6 名用户就足够了。这是因为：

- 哪怕是 100 名用户，也不可能找出所有的问题。10 名以上的用户测试将会耗费更多的时间，相应的也提高了测试的成本。而且时间、精力和金钱的过度付出将很难保证我们再次进行下一轮测试。
- 如果有 5 名用户参与测试，他们可能会发现 80% 的问题。在你修正了这些问题之后，他们在下一轮测试中将发现剩下 20% 里面的 80%。第三轮测试将继续这个比例。这是一道很简单的数学题。
- 没有过多用户参加的测试周期短，有助于在一天内完成并总结。你能够马上利用刚刚得到的结果——趁它们还比较新鲜。如果一次测试更多的用户，你将得到更多的笔记，没有时

间处理。

绝大多数的可用性测试都是定性研究，而不是定量研究。定量研究需要相当大的样本量才能达到一定的可信度和有效度，它涉及测量，表现为数值、表格或图形，用于统计的目的。例如，你可以通过定量研究得知全国有多少比例的人是天天上网。

而定性研究得到的只是描述性的结论。只要在用户测试中有一部分人发现了某个问题，那么这个问题就应当解决。

我在这里总结出几个关于测试用户数量的常见问题，它们很能说明情况。

- **我能不能争取一次就找到所有的可用性问题？**

很遗憾，你不可能在一轮测试中发现所有的问题。问题不是站在那里等你去发现的。它们存在于一定的情境当中，而且有些问题会“埋藏”在另一些问题的下面。

例如，要想完成任务 C，用户必须先完成任务 A。但是用户未能顺利完成任务 A，怎么办？要么他在引导人的帮助下完成任务 A 后进入任务 C，要么他不做任务 A，直接被引导人带到任务 C。而这两种方式都会影响任务 C 的真实情况，而错过它存在的问题。所以，要想发现任务 C 存在的问题，就必须得先解决任务 A 中存在的问题。

- **多几个人参与测试，会不会就能发现更多的问题？**

还是很遗憾，你发现不了更多的问题。在测试完 5 到 6 个用户之后，你将发现只是在看着更多的用户在同样的位置出现同样的错误。这个时候人们发现重要问题的几率已经比较低了，或者说测试的效率已经比较低——记住，有些问题是掩藏在其他问题之下的，必须先解决某些问题才能发现更多的问题。

- **多些人参与测试，我是否能发现问题出现的几率？**

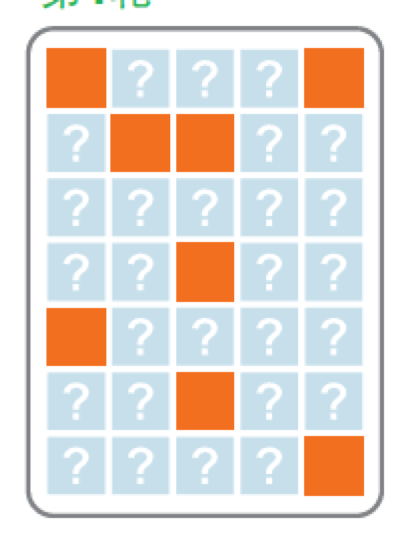
之前已经提到，可用性测试作为定性研究无法得出这样的推论，而且进行可用性测试并不是为了得到什么几率。关键是，在 4 个人出现了同样的问题后，还有必要去知道有多少比例的人会在这里犯错吗？问题已经出现了，你需要做的就是想办法解决这个问题，而不是统计有多少人犯了这个错。

- **我的时间少，资源有限，只测试三个人行不行？**

可用性测试就是要发现并解决问题。测试三个人能不能发现问题？可以。那么，尽管去测试这三个人好了。3 个人的测试和 6 个、8 个人的测试，仅仅是发现问题的数量上的差异。我们不仅赞成，而且非常鼓励这样规模小，周期短的测试。但是有一条，别只测试这一次。

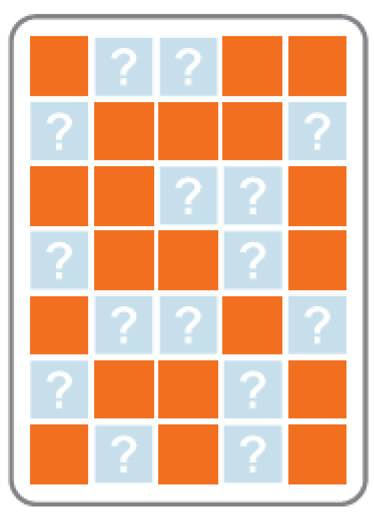
一次测试8到10位用户

第1轮



如果有8名用户，也许在一开始他们发现的问题会稍微多一点。
但是某些问题不解决其他问题总是发现不了。

第4轮

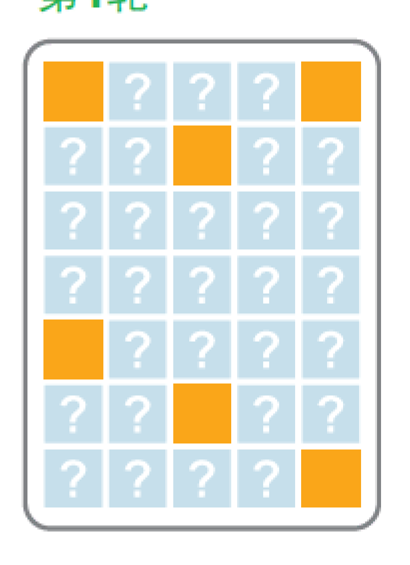


假设每位用户的报酬是150元，那么进行4次测试的费用需要：

(8×150)×4=4800元

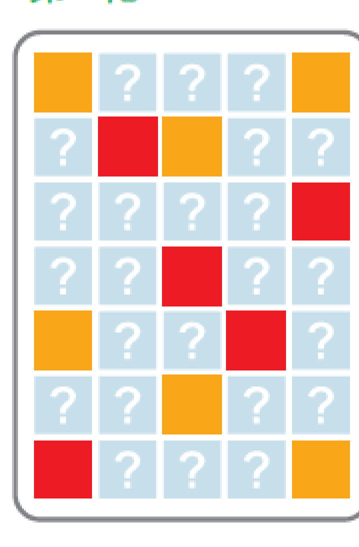
一次测试3位用户

第1轮



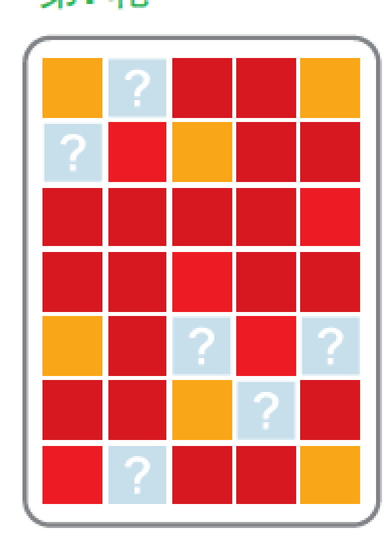
3名用户一开始可能不会发现那么多问题。

第2轮



但如果解决了上次发现的问题后，他们便能够注意到其他问题。

第7轮



进行7次测试的费用需要：

(3×150)×7=3150元

不难看出，进行 7 次 3 位用户的测试，不仅比 4 次 8 位用户测试的开销小，而且能发现更多的问题。

我的结论是，关于测试用户数量的问题，解答的关键不在于某一次需要几个人，而在于是否能够多进行几次测试。你应该争取在每次发现问题并修正之后，再来一次测试，就能发现更多的问题。

换句话说，不管你一次能招募几个用户进行测试，前提是必须保证不会因此而减少测试的次数。测试用户的数量可以取决于你的预算经费，但是测试的次数则不行。

如何招募用户？

我们现在知道了测试用户的大致范围以及数量要求。那么如何招募他们呢？

那些对于用户没有特殊要求的产品，例如门户网站，或者某个在线文字处理系统，我们可以邀请身边的同事、邻居或亲友参与可用性测试，因为他们能够代表大多数目标用户。也

可以说，这些产品所面向的就是所有人，在很多情况下，你既要满足专家的要求，也要满足新手的要求。如果你的母亲也能使用的话，专家就能使用。专家不会介意对初学者来说很清楚的界面。

但是，如果要测试的产品几乎只由某一类用户使用（例如某个股票在线交易网站，或者防火控制管理系统），你就必须认真考虑如何才能找到这些真正的用户。

招募这种测试参与者的办法大致有两种：通过专业咨询机构，或者建立自己的用户数据库。

- **通过专业机构招募**

这里指的是人才交流中心和猎头公司。这些机构都会有自己的人才数据库，他们的网站也能帮助很多公司招募临时或兼职的职员。只要你给出参与者的背景要求，他们很快就能找到相应的人。

但是，这些机构通常是为公司寻找雇员而设立的，所以如果对测试者的要求过于特别，这种寻找就会不太方便。例如他们可以找到某个在线视频网站的用户，但是要找一些曾经使用过你 1.0 版本的人就很困难。

另外，如果背景要求跟工作经验没有多大关系，这些机构也会无能为力。例如我们的在线游戏网站需要测试者不光英语好，还得玩过桌面 RPG。

- **或者自己来干**

另一种方式就是公司建立自己的用户数据库。这样会更贴近公司产品自身的特点。

这种方法就是，列出对测试者的背景要求，例如年龄、计算机的使用年限、对产品的熟悉度等等，然后把它们转化为一个问卷，通过电子邮件或其他形式分发出去。

放在公司网站上供人在线回答也是个好办法。在微软、IBM 和 Adobe 公司的网站上我们都能很容易地找到类似的可用性参加者问卷。

通过回收问卷收集到的资料就能组成用户数据库。既然主动回答了问卷，就代表很感兴趣，那么邀请他们就不会有问题。

找到了用户后，就可以向他们发出测试的邀请。我在这里提供一份招募测试用户的电子邮件样本，你可以根据自己的需要进行修改。

可用性测试的用户招募邀请

在________公司，我们致力于提供给广大用户有用、易学的产品。为了达到这个目标，我们需要您的任何建议和反馈。

我们需要不同经验和技术的人，所以千万不要认为只有计算机专家或是使用过我们产品的人才能参加。这很容易。只要您在白天或傍晚有一两个小时，您就可以到我们位于________的办公室来试用我们的产品。当我们有这样的机会时，将会提前通知您。为了感谢您的测试，您将会收到一定数量的现金以及印有我们公司商标的T恤衫、棒球帽或其他纪念品。

如果您对参加试用我们产品的研究感兴趣，请回复当前的电子邮件，并填写所附的问卷。我们将对您的个人资料绝对保密。

您也可以把这封邮件转发给任何可能感兴趣的朋友。

非常感谢！
________公司可用性实验室全体同仁

姓名: ____________________　　联系电话: ____________________
住址: ____________________　　电子邮箱: ____________________
邮编: ____________________　　出生年份: ____________________

请正确回答下列问题:

1. 请标明您的性别
□男　□女

2. 您的第一职业是什么？
____

3. 您所接受的最高教育程度
□初中　□高中　□大学或大专　□硕士　□博士

4. 您开始使用计算机多久了？
□小于6个月　□6个月到1年　□1年以上

5. 您平时多长时间使用一次计算机？
□每天　□每周　□每月　□每年

6. 请列出您最常用的软件
____

7. 您使用过类似×××的产品（或网站）吗?
□是　□否

8. 您平时多长时间使用一次这种产品（或网站）？
□每天　□每周　□每月　□每年

9. 如果这个产品（或网站）有些令人不满意的地方，您觉得是在哪里?
____

8.3.3　主持人和观众

有些人觉得可用性测试很高深莫测。需要非常有经验的可用性测试专家，在测试的过程中言

语睿智，把握大局，然后在一周之后产生一份 20 页的专业测试报告。

也许测试的引导人确实需要一定的技术和常识，但别把这个问题想得那么神秘。

谁适合当测试的引导人？

一个合格的引导人都需要做些什么？

- **大多数情况下，什么也别做**

在测试的过程中，不要过多的干扰用户，让他们在尽量真实的状态下使用。不知道你怎么样，反正我使用某个产品的时候旁边没有人指指点点。

很多初次引导测试的人，看到用户犯错就很热心的去帮助，这其实正犯了测试的大忌。你的目的是为了发现问题，那么你要做的事情就是等待问题发生。你需要看到的是用户犯错的频率、解决问题的办法、他们的表情、出现错误后的行为。有时候从用户的某些举动还能看出他希望什么。这些观察可以在很大程度上帮助你改进设计。

- **在恰当的时候引导用户去完成几个任务**

有些用户完全是新手，或者过于紧张，试用产品的时候完全没有章法。别因为他的一无是处就放弃这次测试。他肯定能发现问题，这种时候你可以适当的引导和提示，帮助他走上正轨。有时候这种胡打乱撞的用户往往能够发现更多的问题。

另外，有些测试任务是事先安排的。引导人可以进行简单的讲解，提示他现在该做什么。

- **测试完成后进行一些必要的交谈**

在完成测试后，和用户进行一些交谈是有必要的。例如你对他的某个行为有疑问，这个时候就可以试探性地沟通一下，以了解他在操作时的真实意图和感受。你也可以和用户聊聊对产品的感觉、期望等等。更深层次的交谈还可能会发现一些被疏忽了的需求。

引导人在测试中所要做的就是这些。你真正需要的工作只是鼓励用户去尝试。只要有一些实践经验，几乎任何人都可以引导可用性测试。

找一个冷静、有耐心、富有认同感的人。他善于倾听，而且天性比较客观和公正。就是这样。只要他不是动不动就拍桌子发火，或者天性喜欢“死磕”，就没问题。我觉得你就完全可以胜任，难道不是吗？



希望这种脾气的人不要主持测试

谁适合当测试的观察者？

任何人都可以观察测试，只要他愿意——不管是设计师、软件工程师，还是市场推广人员，甚至是公司老总。

或许测试的工作是由你来安排，但测试绝不是你一个人的事。为了保证团队统一认识，以及接下来的改进工作能够顺利开展，有必要让所有人一同认识到产品存在的问题。

而且，很可能你在观察中的意识是有局限的。更多的人参与观察，能够从更多层次来发现问题所在。例如，有些产品反映速度的问题，并不一定是用户界面设计所能解决的，也可能是服务器某个配置不合理的问题——有时候你需要从其他方面来考虑修改工作。

最好让你的团队领导或者管理层一起参与测试。大胆地邀请CEO，告诉他，如果他和高层经理们能参与几分钟，将会极大地调动开发团队的积极性，他们会很乐意参加。而一旦他们走进了观察室，我想他们肯定会呆上不止几分钟。因为这是他们第一次看到自己的产品在被人使用，而且很多地方和他们想象的美好画面并不一样。

高层经理们在参与了测试后会体会到可用性测试的重要性。这也便于让他们以后更加支持可用性测试，并且愿意增拨一些研究经费。何乐而不为?



你凭什么要求我们继续为实验室拨钱?

8.3.4 道德问题

可用性测试可能需要参与者的个人资料，例如姓名、健康状况、雇用情况、教育程度、居所和财务状况等等。有时候这些东西他们并不希望公开。而另一方面，公司测试的产品信息也应该是保密的，以防止可能的竞争对手获得相关的情报。

这种道德问题是双方的，所以最好制定一份协议，首先让用户放心，同时也保护自己，避免不愉快的诉讼行为以及泄漏商业机密。

下面是一份知情协议书的样本。

参加者编号：__________

了解你所参与的测试

你的名字：__________
请仔细阅读以下条款：

测试目的

本次产品测试需要你的参与，目的是为了帮助我们测试我们的产品是否简单、易学、好用。这个产品测试是研究你将要用到的产品，而不是在测试你，或是你的能力。

测试收集的数据

测试将会记录你是怎样使用本产品的，例如本产品的某一个部分是否简单易用。在测试中你将会回答一些问题，可能在测试后还会有一个简单的口头调查。

实验中你所提供的信息，以及其他参加者的信息，将帮助我们找出改进本产品的方法。

同意和弃权声明

你使用本产品和口头调查的过程可能会被录像或录音。我们公司在日后的评估或演示中将可能使用你的声音和录制的图像。但我们不会公开你的姓名。在这份协议上签名，将表明你同意以上问题。

保密声明

在测试中你所得到的任何关于本产品的信息，都是保密的，并归我公司所有。你在测试中得到的信息仅仅是为了试验的目的。在这份协议上签名，将表明你同意保守秘密，不将产品的信息泄露出去。

酬劳

你可能会参与3到8轮测试，每一轮测试的酬劳是__________。在这份协议上签名，将表明你同意酬金的数目，不会在以后提出异议。

舒适

如果需要，你在实验的任何时候都可以要求暂停，你只需要告诉工作人员你要求暂停就可以了。

> 退出测试的自由
>
> 你参加本次测试完全出于自愿。你可以在任何时候退出测试。
>
> 如果你同意以上条款，请在下面签字。
>
> 签字：________________　　日期：________________

协议应该写得尽量简短，而且通俗易懂。也许上面这份协议样本过于 嗦和正式了，但是我把它放在这里是为了让你知道有些问题需要考虑得全面一些。你完全可以对它再进行省略、修改或者加工。

8.4　测试的过程

下面我们看看可用性测试的全过程。

8.4.1　设计好考题

每一次可用性测试都会发现问题，这些问题可能是各种各样的。有可能是界面比拟不太合适，或者是层次结构不清晰，或者没有贯彻一致性，也有可能是文字辞不达意。与此同时，作为开发者内部也会存在一些有争议的问题，或者一些疑惑，需要用户在测试中给出结论。也就是说，测试之前你会有一些目标。

这些目标将会影响到测试的具体方法，以及测试任务的设计。让用户完成某项任务是最经常的测试方法。这些任务应该是典型的，也就是能代表大多数用户将会经常碰到的关键步骤。这个时候我们之前做的“任务描述”（看看 2.4 节）会很有用，因为当时考虑的都是典型任务。

下面是设计任务时要注意的几个方面。

- **提高用户的积极性**

  给用户真正的任务，而且让他们有权选择。如果是测试在线购物网站，给用户 200 元现金让他去买自己想要的东西，并把东西寄给他自己，这就比“假设你现在要买一张 40 元以下的 CD，然后寄到 ××× 去”要好。当人们执行苍白、呆板的任务时，很难投入情感，也不会尽可能地运用个人知识。

- **逐步加大难度**

  把任务的难度以一种阶梯形式排列。第一个任务不妨设计得简单一些，这样会帮助用户尽快熟悉测试的过程，建立自己的信心，并且他们能够很快平静下来，专心完成下面的主要任务。



好的，第一个任务是找到骨头，
然后再看看如何搭建狗屋。

8.4.2 介绍阶段

在测试开始前，你应该花一些时间向用户介绍测试的过程和设备，以及为什么让他来这里。

工作人员轻松和蔼的态度能减轻用户的压力。为了帮助他们放松下来，在开始测试之前闲聊几句题外话会很有效果。这能极大地松懈他们的紧张情绪，而且能让他们多说话。测试的时候需要他们说话。

有时候你还能从和用户的闲聊中了解到他或她是个什么样的人，对相关产品的熟悉程度。例如，我可以从他从事什么工作聊到他每个星期会花多长时间上网。也许一开始用户只会联想到工作，例如“收发电子邮件”或者“用 MSN 与客户沟通”，但随着话题的打开他没准会想到自己还曾经在淘宝上为女友买过一些礼品。如果我们正在测试的是一个在线购物系统，那么就能得到有用的信息——这不是一个新手。

为了消除参与测试用户的紧张情绪，以下这些话都很有必要 ：

首先，放轻松，我们是测试这个网站，而不是你本人。所以你不必担心会犯什么错误。

其次，如果你觉得有些地方让你感觉不好，只管告诉我。我们本来就是想知道你是怎么想的。

第三，如果有问题就尽管问。有些问题我可能不能马上回答，因为我们想知道如果没有别人在旁边的时候，人们会怎么做。但在测试结束后，我会设法解释任何你还有疑问的地方。

最后，如果你感觉有点累，我们可以喝杯咖啡，休息一下。当你感觉不想继续的时候，你也可以告诉我。

相信我，一些简单的饮料或者小点心会非常有效——哪怕他们并不需要。

告诉他有个摄像头，但是别因此而紧张，也许他的表情夸张一些反而更好。这时你需要给他一份知情协议书。

8.4.3　正式测试

介绍环节过后就开始正式的测试了。正式的测试分为两个主要阶段。

- **“理解”阶段**

让用户看到你所要测试的网站、Web-based 产品或软件。询问他们对页面的初始印象，然后看他们能否理解产品的大致功能、层次架构和运行方式等等。

- **“任务”阶段**

让用户完成一些任务，观察他们在任务过程中的表现，记录发现的问题。

在这两个阶段中，你需要注意以下几点，便于正确获得所需要的信息。

观察用户

引导人可以面对面地直接观察用户，观察室内的其他成员则可以通过摄像机或者单向镜子观察。

观察可以了解用户的情感反应，例如叹气、皱眉、耸肩等等行为，这些体现了用户的不满和受挫情绪。有时候他们表现出的惊喜并不一定就代表着肯定，因为这种“突然就捡到的果子”表明并不是他们预料中的结果，他们是无意中瞎打误撞出来的，就好像“瞎猫碰见死耗子”。



这很简单，就像在看电视。
有时候我还能吃点零食……

除了表情你还需要观察用户鼠标的移动和点击。如果鼠标漫无目的地在页面中移动，或者想点击某个按钮又不敢点击，就表明他们对自己的操作没有信心。

但是用户有时候会因为正在“被观察”而感到紧张。单单知道那面镜子后面可能有一群人在看着他，用户也会感到不自然（女性尤其如此）。这是一个无法确定的因素。而且单纯的观察会遇到一个问题，就是你有时候并不太确定他们在想什么，而只能根据观察到的现象揣测。

让用户进行有声思考

我们需要知道用户的想法和思维过程。如果他们在测试过程中保持沉默，我们无法猜出来发生了什么，或者他们在思考什么。

为了避免这种情况，你可以鼓励用户口述自己在想些什么。他可以用语言表达他对系统的理解，喜欢或不喜欢哪些方面，以及描述自己为什么要采取某个行动，期待的后果是什么。通过用户的讲述，我们可以了解他的认识哪些是正确的，哪些又是误解。我们也能知道哪些地方引起了他的误解。

这种有声思考例如“你想让我买书？那么我想我应该点击‘图书’或者‘书籍’栏目……为什么我找不到？”、“用户名和密码？我买本书难道还需要注册？”或者“我觉得这里的广告动画很晃眼，它让我无法集中精力”。

有些人可能会感觉不停地自言自语显得有点傻，或者尴尬、难为情。有时候他们也会忘记需要说出想法，因为我们都不习惯于自言自语。你也可能会发现，当任务难度增加时，你很难边说边做。实际上，在执行困难任务时，人们往往会停止自言自语，而这恰恰是我们最想知道他想法的时候。

这类的“沉默”是有声思考面临的最大问题之一。

当用户沉默时，我们可以用类似“你现在正在想什么？”的话来提醒他们继续口述。另一种方法是让两位用户共同合作，他们可以互相讨论、互相帮助。这种合作方式会更加自然，而且能揭示许多信息。

提出问题

有时候我们会向用户提问以了解或者核实我们想要的信息。问题应当简短、明确，而且应避免询问过多的问题。

关于询问用户你需要注意几个地方。

- 不要问过长的问题，因为它们不便于记忆。
- 不要问太复杂或模糊的问题。把它们分解成几个独立的问题会更合适。例如，“这个网站和其他网站相比，你觉得如何？”显然就太宽泛了，用户也不知道该从哪里说起。而把问题变成“你觉得这个网站怎么样？你觉得有什么其他网站与它比较类似？你觉得那个网站怎么样？”就会更容易回答，也便于做记录。
- 不要使用可能让用户感觉尴尬或无法理解的术语。如果你提的问题用户不懂，他的自信会受到打击。而且你还要解释，这会浪费时间。
- 不要问带诱导性的问题。例如“你觉得这个网站好在哪里？”或者“你喜欢这种颜色，对吗？”就带有一种假设，即用户觉得它不错。同时也不要把你自身的偏见带入问题。

尊重测试用户

测试的用户可能会感到很大的压力。想想看，陌生的产品，陌生的引导人，在镜子或摄像头另一端还有一堆人不知道是不是在等着看笑话。如果用户在测试中犯了一些错误，或是在学习的过程中显得有些慢，他们总是会觉得自己愚蠢或是力不从心。

千万不要表现出不耐烦，甚至讥笑他们的表现。始终和颜悦色是必须的，并且让他们感觉到任何失误都没什么大不了——本来也正是如此。

另外，在测试临近结束的时候，你可以安排一些相对容易的任务或问题，这样便于消除用户的紧张感觉，而不是以一个失败者的角色结束测试。

如果测试超过 45 分钟或一个小时，应该给用户中间休息的机会。也应该让他们知道自己随时有权中止测试。

一个实例你就能明白

我正在为“招商证券”的网站进行一次可用性测试，可能会对它的改版提出一些评估意见。参加测试的这个用户有一些炒股经验，但没有使用过这个网站。



测试已经进行到了一定阶段，以下是我和他的对话（蓝色是我说的，黄绿色是用户说的）：

"好，现在我们来看看别的。你觉得这个网站还能帮你做些什么？"

"我刚才留意到'软件下载'栏目。我想我可能会找到某个软件，它能在我的 Pocket PC 上运行。很多证券网站都提供类似的软件，这样我就能随时随地了解股市行情了。"

"很好。那么你会怎么做？"

"我会点击这个链接。"（他把鼠标移动到页面的右上方）

他点击"软件下载"的超链接，打开了一个新的页面。他看了很久。



“有什么问题吗？”

“这个网站没有针对手机的软件。”

“为什么？”

“下载列表里面的软件不多，而且都是电脑上使用的。”

“确实没有吗？”

“确实没有。”

他非常肯定。现在测试已经走到了死胡同。但是任务显然并没有完成，而且我感觉隐藏的问题还没有完全被挖掘出来。于是我决定给他一点提示。

“你注意到右下方那几个图片按钮了吗？有红色字的？”

（阅读）“……噢，在这儿。我还以为是广告呢！”（很有信心地点击）

这时弹出了一个新页面：



（鼠标指向左上角的页面logo）"'手机炒股'，没错，就是这里，我想我找到了。"

"那么接下来？"

（鼠标在左边的页面导航区域缓缓移动）"我想我得点击……"（看着"移动证券"和"掌上股市"两个链接在犹豫）



"我不太确定该点哪一个。这两个好像都有点像，它们都在说我能随身炒股，我不知道这里有什么区别。"

"你感觉哪一个最可能是你想要的？"

"……'掌上股市'，我想。因为我就是拿来在掌上使用。"

"点它试试看？"

他点击"掌上股市"后出现的页面：



“没错，就是它！（阅读右方主窗口内文字）……这是给 CDMA 手机使用的软件……我的手机不是 CDMA 的……也许我应该点‘移动证券’。”

点击“移动证券”后出现的页面：



（阅读）“这好像只是个短信业务，不是软件。”

他再次点回到“掌上股市”页面，更加仔细地浏览，但是没有收获。我发现他已经开始焦躁不安。

“有什么新发现吗？”

“没有。我觉得这个网站不提供我想要的服务。”

测试到这里告一段落。他没有找到他想要的软件，于是认为网站并没有提供全面的服务。但是

他错了。网站有他想要的软件，只不过不是在“移动证券”和“掌上股市”中。如果他注意到了上面的那个“招商移动股市”链接，可能就会马上完成任务。但是似乎这个链接并不吸引浏览者的注意力。



到这个时候我可能会直接宣布结束测试，或者再布置一个简单点的任务，让他能从消沉的心态中放松。适当的安抚是有必要的，不要让用户为没有一个好的测试结果而耿耿于怀，因为有可能过一段时间仍会让他参与后续的测试工作。

8.4.4 测试之后马上回顾和总结

记住，在每轮测试之后，你应该及时让开发团队交换一下意见和认识，回顾每个人的观察，并总结记录。这会让问题得到更清晰、更有条理的呈现。然后你应该决定接下来该如何处理这些问题。

如果测试后不及时进行总结，时间拖久了你就会丢失一些原本很好的想法，或者减弱某些感受，更糟糕的是你会遗忘某些感受。如果是这样的话，测试就不那么有效果了。

在每轮测试之后，我们会收集到一定数量的数据，它们通常分为两类。

- **客观数据**

  客观的数据包括一些加法题，例如用户完成任务所需的时间、犯错误的次数、用户接受帮助的次数等等；同时还有一些是非题，例如用户是否能独立完成任务、是否使用了最简捷的途径。

- **用户的主观感受**

  这些信息需要观察者分析用户表现出来的一些现象。例如用户对产品的印象和评价、用户在哪些地方感到迷惑、注意力是否被转移等等。

此外你也可以在测试后请用户总结他对产品的感觉。提供一些简短的试验后问卷能够反映用户对测试产品的满意度。

这里也有一份测试后满意度问卷的样本。

参加者编号：___________

你对这个产品的感受

这个问卷给你一个机会来告诉我们你对刚刚使用该产品的感受。

了解你对产品感到满意和不满意的方面，能够帮助我们改进设计。

在回答每一个问题的时候，请尽量回忆你完成的所有任务，然后选择最能表达你感受的答案。如果某个问题无法回答，请选择“无法回答”。

1．这个产品使用起来很简单

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

2．我可以用这个产品顺利地完成任务

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

3．我能够使用这个产品较快地完成任务

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

4．使用这个产品时，我感觉舒适

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

5．我认为使用这个产品可以帮助我提高工作效率

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

6．这个产品给出的错误信息能够清楚地告诉我发生了什么

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

7．当我犯错误的时候，我能很容易地迅速从错误中恢复

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

8．这个产品的文字描述非常清楚

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

9．这个产品的信息很有帮助

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

10．我能很容易地找到我所需要的信息

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

11．这个产品的信息在屏幕上组织得比较清晰

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

12．这个产品的用户界面让人感觉舒适

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

13．这个产品提供了所有我期望的功能

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

14．总体来说，我对这个产品感到满意

非常不同意<————————————————————————>非常同意

1 2 3 4 5 6 7 无法回答

8.5 让专家来评估（别忘了，你也是专家）

有时候寻找用户并不容易，让用户参加测试也可能需要过高的成本和时间。在这种情况下，你也可以考虑聘请一到两位可用性专家来进行评估工作。

当然了，如果你请不起专家，或者不知道怎么请专家，没关系。这本书看到这里，我相信你也能够对可用性说出个一二三来。在没有专家的情况下，你就是专家。这没有什么不好意思的。

8.5.1 现成的原则——启发式评估

这种对可用性的评估不需要可用性实验室，也不需要引导人或者观察者。专家（或者你自己）根据一些常见的可用性原则来评估产品，通过自己的经验来发现潜在的可用性问题。这其中最有代表性的就是 Jakob Nielsen 博士在 1990 年提出的“启发式评估法”（后来他又进行了改进和确认）。

启发式评估并不难，它只是把一些具体的可用性原则放在我们面前，然后让我们一条条地去核对产品是否满足这些要求。

这些可用性原则我相信你已经很熟悉了——“摩西的十诫”。在第一章我们就提到过，第三章的物理设计指南里面也提到过。不过我还是在这里再次把它们列举出来，因为它们真的非常重要，尤其是当你自己来评估的时候。

- 让用户随时了解系统的状态
- 系统应与真实世界相符合
- 给予用户控制权和自主权
- 提倡一致性和标准化
- 帮助用户识别、诊断和修复错误
- 预防错误
- 依赖识别而非记忆
- 强调使用的灵活性及有效性
- 最小化设计
- 提供帮助及文档

上述这十条是针对所有交互式软件产品的。如果你设计的是商业网站，可以看看 Nielsen 提出的另一组名为“HOMERUN”的启发式原则。

- 高质量的内容（High-quality content）
- 实时更新、经常更新（Often updated）
- 减少下载时间（Minimal download time）
- 方便浏览者使用（Ease of use）
- 符合用户的需要（Relevant to users’needs）
- 在线媒体的唯一性（Unique to the online medium）
- 网络化的企业文化（Netcentric corporate culture）

至于选用哪一种原则，或者哪些原则可以稍稍做出让步，这取决于你和开发团队的共同意见。记住，规则都是死的，我们要保障可用性，同时产品也要有自己的个性。

8.5.2 你也可以看看这些具体的评估角度

上面那些原则你应该很容易就能够理解。我在这里也提出一些具体的评估角度，这样你就能更方便地检查自己的产品是否能符合下列这些要求（通常来说，满足这些要求就够了）。

关于一致性的评估

不管是 Web 还是 Web-based 产品，一致性都是用户界面最基本的要求。良好的一致性能使用户在使用的时候很快就熟悉产品的操作环境，同时也能避免对相关操作的理解产生歧义。记住，用户界面是一个整体。

你的产品是否具有一致性？看看它是否满足以下几个方面：

- 所有界面在外观、布局上是否一致？
  例如，网站的每个页面是不是都有导航栏和 logo 标识？
- 功能类似的相关界面，在外观、布局以及交互方式上是否一致？
  当你在网络书店搜索图书的时候，通过书名搜索和通过作者名搜索，给你的感觉是否一致？
- 当存在系列产品的时候，这个产品与其他产品在外观、布局以及交互方式上是否一致？
  想想微软的 Office 系列，还有以前的 Macromedia“网页三剑客”。

- **在同样的环境下，同一层次的文字在字体、大小、颜色、对齐方式上是否一致？**
  一般情况下的文字是不是一致？ 每个弹出窗口里的文字是否一致？每个表格里的文字是否一致？

- **导航栏或者菜单的格式是否一致？**
  你应该不会在一个网站里，把有的导航做成下拉式，有的导航又做成标签式吧？

- **各个控件之间的间隔和对齐方式是否一致？**
  如果在大多数情况下，控件是左对齐，那么一般来说控件最好都左对齐。当然，如果你觉得在某个极端情况下，右对齐会更好，那么就右对齐。但只要是类似这种情况，就必须都得右对齐。

- **提示、菜单、帮助中使用到的术语是否一致？**
  尽量不要使用术语。如果有术语，那么术语就应当一致。

关于界面简洁性的评估

你的用户界面像你的脸一样对称、干净吗？我看过一些用户界面设计就像长了天花。因此你不得不对它进行美容前的检查。下面是一些供检查的建议条款（实际上，关于界面简洁性的一个评估原则就是：1. 干掉多余的东西；2. 尽可能分组）。

- **用户界面是否存在空白空间？**
  虽说界面不能太散，但空白是必须的。没有空白空间的界面叫做“杂乱无章”，可用性极差。

- **用户能否很快找到他所要的信息？**
  不要把本来应该展现给用户的内容隐藏到五花八门的广告里面。

- **菜单的深度是否在三层以内？**
  我建议菜单不要超出三层。如果层次太深，将不便于用户记忆，而且他们要到达某个地方会很麻烦。

- **界面的控件分布是否按照功能进行了分组？**
  菜单就是菜单，工具栏就是工具栏。同样的，单选框组和复选框组也得分开。

- **一个类型的数据是否需要通过移动滚动条才能显示完全？**
  如果是这样，尽量采用分页显示，并且提供数据排序显示的功能。

关于信息反馈的评估

假设产品的用户是一个初出茅庐的新手，你能指望他在操作的时候不出错吗？但这还不是问

题的所在。问题在于谁都会犯错误，我们都有自己不了解的东西。

要想避免这类问题，就要求用户界面有足够的输入检查和错误提示功能。通过信息反馈，用户得到出错的提示或是任务完成的赞许。不幸的是，很多产品在这方面都做得不尽人意。下面是这类评估的一些参考意见。

- **系统是否在用户输入之前，就给出具体输入方式的提示？**

  前两年我注册过很多网站会员，但很少能一次就顺利成功。我总会犯些错误。现在好多了，他们会事先告诉我“用户名只能使用字母和下划线”。但更好的是提供多个文本框，避免我出现格式或语法错误。

- **系统是否接受用户的正确输入，并做出提示？**

  例如在用户点击提交或上传按钮后，应该提示他系统正在处理或者显示上传进度，而不是让他干看着之前的页面不知道进展如何。

- **系统是否拒绝用户的错误输入，并做出提示？**

  如果用户做错了，就应该拒绝他的操作。但必须让他明白自己错了。我们可以通过弹出警告框或者发出警示音来提示他。

- **有没有操作成功的提示？**

  很多系统都缺少这一步，使用户毫无成就感。

- **如果是提示他们错误，那么提示是否正确并且浅显易懂？**

  他们出错本来就很晕，这时候千万别拿“ERR0083”之类的玩意来让他们更摸不着头脑。

- **提示所用的图标或图形是否具有代表性和警示性？**

  使用正确的图标，用户的感受会加强很多。

- **系统提示的用语是否按警告级别和完成程度进行分级？**

  只要不是故意的破坏性操作，一般最好对用户温和一些。

- **用户界面是否提供了突显功能？**

  例如按钮或超链接，鼠标移动上去会不会有相应变化？

关于用户动作性的评估

“科学是懒人的哲学”，有时我很认同这个观点。我们的系统能让用户尽可能地偷懒吗？例如少动些手肘，少记些命令？没有一个用户愿意做得多而收获得少，就好像去郊区一天，结果只钓到一条鱼一样。

记住，用户从某种角度上来说都是心怀不测的挑衅者和肇事者。当他们对待自己寄以很大期望的系统时，很少会有太多的耐心。下面是一些判断用户是否能够“偷懒”的方法。

- 是否存在更便捷、直观的界面显示方式？
  尝试着多想想。如果有，换过来。

- 是否存在快捷键以代替用户最频繁的操作？
  例如大家印象最深刻的“Ctrl+S”，还有用“Tab”键切换文本输入框。

- 是否支持对错误操作进行可逆性处理，返回原有状态？
  例如“Undo”和“Redo”指令。如果视频上传到一半的时候发现是不健康的内容，别告诉我不能取消……

- 系统的反应速度是否符合用户的期望值？
  系统反应太快或太慢都不好：太快了吓人，太慢了磨人。

- 用户在任何时候，是否都能找到并开启帮助文档？
  例如绝大多数软件中的 F1 键。

- 系统能否减轻用户的记忆负担？
  如果菜单或导航的内容太多太复杂，我就只能靠自己的脑子来记住常用选项的位置。

- 是否采用了相关控件来替代用户手工键盘输入？
  例如日历控件和计算器。

- 不能执行的操作是否被禁止？
  如果这个按钮不能点击，那么请把它置灰。

- 能否方便用户的选择操作？
  例如选项过多的情况下，是否可以采用下拉列表来显示？

关于产品特色的评估

每一个 Web 或 Web-based 产品也都应具有自己的个性，从而体现出开发商（也就是你所在的公司）的特色，同时这也是针对用户群体的特定需要。例如微软和苹果的界面风格就完全是两码事。

一个具有个性的产品，它本身就是开发商的“广告代言人”。但需要注意的是，我们既要突出

开发商，又不能喧宾夺主。下面我们给出一些常见的软件特色评估的原则。

- **网站或产品的界面是否有单位介绍或产品介绍，并拥有自己的图标？**
  例如网站的 logo 标识是否得体？在线软件的登录界面是否有开发商信息？

- **是否有版本查看机制？版本说明上是否有制作者的标识？**

- **产品界面的色彩、背景、布置是否与同类产品有不同之处？如果有，是否更为简洁、美观？**
  这些就是产品的直观特色。

- **产品界面的操作与同类产品相比，是否更简单快捷？是否更能预防错误或提示错误？**
  这些是产品的非直观特色。

- **产品是否为特殊群体或是特殊的应用提供相应的操作机制？**
  例如 Windows 的“放大镜”工具，或者网页中的设置字体大小功能。

总之，对用户界面的评估需要一个立足“共性”但又要强调“个性”的评估思路。对用户界面的评估更加强调从用户的角度和审美观去看待产品，既不能过于“大俗”，又不能过于“大雅”。

很多时候，你需要在强调规整和强调个性间进行权衡。这需要你用大脑去思考，用心去体会。对你的审美观也是一个极大的挑战。

8.6 测试需要修正

可用性测试后当然需要对发现的问题进行修正，否则测试还有什么意义？

修正应当及时、准确。当修正之后，还需要再次进行用户测试对其加以检验。但我们也需要注意以下几点问题：

有些问题可以忽略

对于有些问题，可能根本就无法修正。这些问题总是存在。

例如在网上购物的时候，总是有人在“电器”栏目里找 iPod，在“电脑配件”里找 GPS。你不能说他们错了，但很明显这么规划栏目又有些不合适。你也许知道 Apple 网站里面有很多电影预告片，但是它们以前一直放在“QuickTime”栏目中。这似乎也不太恰当，但没有其他更适合的栏目。

对于这些问题，忽略它。无论怎么研究，可能也找不到完美的解决方案。记住用户也在不停

地学习。只要这种学习过程并不令人难以接受，并且容易记忆，那就没问题。

有些要求也可以忽略

有些用户在测试的时候总是说：“如果它有这个功能就更好了。”这样的说法常常被理解为他们在要求新的功能。

但这并不是在告诉你，如果没有这些功能，他们就会真的觉得这个产品烂透了。反过来说，就算你加了这些功能，他们也并不一定就会跳起来为你叫好。

如果你仔细询问常常会发现，说这些话的用户已经找到一个很好的网站（或者他们知道某个产品）有这个功能，他们并不会因为你加了这个功能，就放弃那个网站或产品而只用你的。同样的道理，你没有加这个功能，他们也不会因此就放弃你提供的其他功能：他们只是在告诉你他们觉得什么很酷而已，这些不是真正的要求。

别顾此失彼！

这是最重要的。做任何事情都需要讲究平衡。这种平衡都是很微妙的，我们必须时刻都进行全盘的考虑。

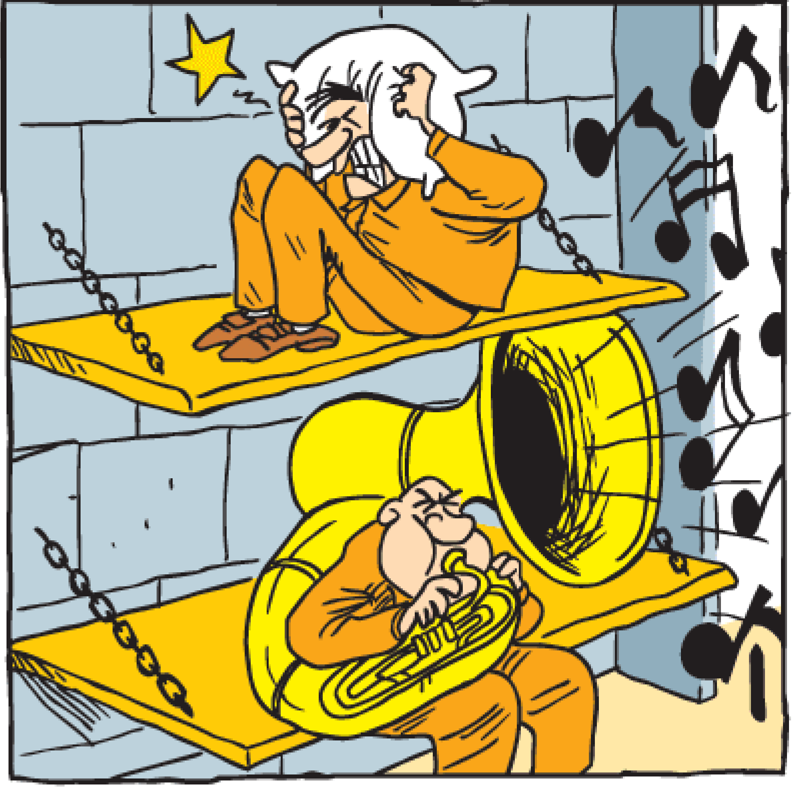
比如说，你发现某个地方不能吸引用户的注意力。为了修正这个问题，你把它调整得更为突出。但是这种时候你必须注意，会不会因此而降低了其他内容的重要性——船头高了，船尾自然就会低下去。

图片也是一个需要权衡的问题：用户觉得网站页面显示得很慢，你就要想方设法提高这个方面的可用性。但如果你希望最快的页面下载速度，那么就一张图片也别放。但这样网站就会很难看。

有时候，哪怕一个很小的变化都会带来很大的影响（想想“蝴蝶效应”）。真正的挑战不是如何修正你发现的问题，而是修正这些问题的同时不破坏已经正常运行的部分。

记住，有时候你的改变会得到一些东西，但也可能会失去更多。就像下面这个可怜的家伙一样。





第九章

了解满意度

Just one last dance…oh baby…just one last dance.

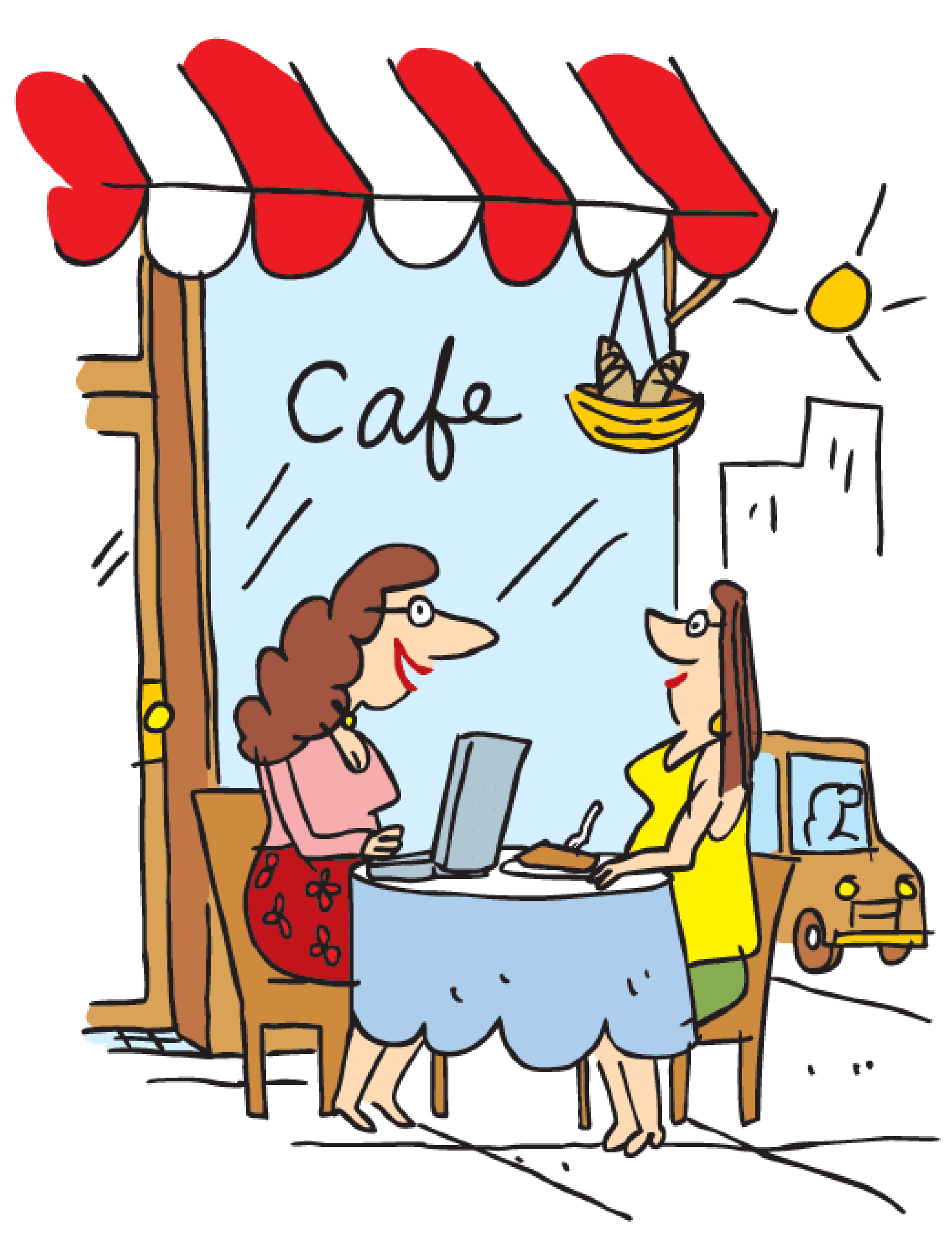
——专辑《Key to My Soul》，歌手Sarah Connor

现在让我们开瓶香槟庆祝一下。从最开始的产品创意，寻找用户，找到需求，然后不断地开发原型并且评估，最后测试产品的可用性。在这些环节之后，产品终于可以面市了。它也许不是最出色的，但肯定不差——因为它经受住了我们的重重考验。

但是产品的推出并不代表着设计结束。别不耐烦，我们已经做了这么多，不妨把这最后一点工作做好——因为它和测试一样重要。这一章也很短，所以请看完它。

这最后的一点工作是什么呢？

记住我们一直在以用户为中心。我们已经给餐厅的顾客们上菜了，但作为负责任的餐馆，这还没完。菜的味道到底如何？客人们的感觉怎么样？我们只能去问顾客自己。



您觉得我们店牛扒的味道怎么样？

也就是说，你还需要收集用户的实际使用情况。有必要的时候，根据这些信息，你也可能会不断地改进和提高现有产品的质量和可用性。

其实这种收集活动非常简单，方法也是我们所熟悉的。关键只在于你愿不愿意去做。

9.1 询问用户——问卷调查

问卷调查用于统计数据，以及搜集用户的意见。它的好处是可以搜集大量用户的意见。通过

电子邮件的形式，用户可以自己决定是不是回答问卷。没有人逼他们。



这就有些荒谬了

而一旦他们愿意回答问卷，那么通常都会很认真地填写。这样你收到无效问卷的几率会少一些。

那些在网站上的小调查几乎没有很大意义，因为问题过于笼统，也不清楚接受调查的用户状况。所以，许多问卷调查都是从询问基本的用户信息开始，例如他是使用产品 5 年的老用户，还是刚刚接触的新手。



这种调查还真是有点莫名其妙

至于问卷的设计，它和我们在测试后的小问卷几乎一模一样，无非是用户对网站或产品的感受。这包括可用性、信息的质量和用户界面的质量三个方面。

9.2　询问用户——访谈

另一种在产品推出后的跟踪调查方法是实地访谈。直接面对面地与用户交流，观察他在实际工作环境中的使用情况（“环境”的重要性），询问他的感受。你会发现很多实际中的问题。

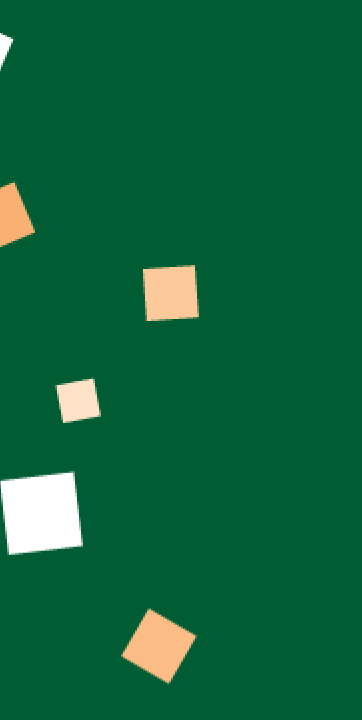
例如，用户在使用产品的过程中可能会受到一些干扰，接到电话或者有同事打断。他们受到干扰后会尝试保持或恢复到干扰前的状态。这些可能都是问卷调查所发现不了的情况。

其实满意度调查可以说是用户测试的一个延伸，它并不是迟到的测试，因为其目的是为了产品的改进，或者是为了下一个版本的设计。这种用户至上的理念，也会博得用户们的好感，他们会觉得你是一个负责任的开发者。由于这能更好地支持他们的工作，他们也会表现得非常合作。



终　章

回到开始：什么是好的用户界面设计



门徒们问耶稣：“告诉我们尽头在哪里。”

耶稣说：“那么，你们已经找到了开始，现在要寻求结束了？开始在何处，尽头就在何处。”

——《托马斯福音》

就这些了，朋友们。

在你阅读这本书的时间里，我们一同经历了“以用户为中心”的用户界面设计全过程，从了解用户、建立用户需求，到设计方案和开发原型，然后正式地设计，并迭代进行用户测试。最后还可以通过一些方式来了解用户对产品的满意度。

同时，我们也了解了文化的差异性，这对基于互联网技术的产品开发是至关重要的。

在第六章和第七章我们还专门谈到了 Web 和 Web-based 产品的可用性。你已经发现很多产品都存在着用户交互方面的问题，而且页面的每个元素都需要加以注意。

干得不错，伙计。我简直有些不敢相信你已经看到了最后一章。能看到这里的朋友，不管以前是否接触过正规的用户界面设计，我想你现在已经理解了这个概念——它并不难，只是需要更科学、全面地看待设计而已。用户界面设计不靠灵感，它需要的是你的勤奋和不懈的实践与验证精神。

什么是好的用户界面设计？现在再问这个问题，我想你已经有了答案。好的用户界面设计就是以用户为中心的设计，可用性强的设计，美观的设计，让用户有良好体验的设计。如果不确定自己的产品是好还是不好，邀请几位用户，让他们体验之后就会得到结论。

其实，没什么难的。看完这本书我想你应该会有这种感觉。如果有权利的话，我会为你颁发毕业证书。但是一个证书并不能代表什么，真正关键的是你是否掌握了这些知识，并能把它们付诸实践。



我在书中描述了很多用户界面设计使用的正规方法，包括建立用户模型（创造你的“阿童木”）、数据搜集（问卷、访谈、观察和集体讨论）、用需求模板来解释和分析数据（一道填空题）、描述任务（讲

故事和流程图），以及让用户参与（师徒关系）等等。

此外还有很多原则和指南，例如“摩西的十诫”、“8 项黄金法则”、“HOMERUN 原则”。这些都非常有用，如果你记不住的话，随时翻翻相关的章节。

但是，就像我说的，我只可能为你颁发毕业证书，而不是学位证书。有时候这是两种东西，就像有些研究生文凭一样。毕业证书只是一种学历证明，学位证书则表示你在某个专业领域已经达到了一定的高度，所以给你的认证。我只能说你学习完了这些知识，却不敢保证你是否能够把它们运用到实际工作中去，并且运用得好。

我讲了很多技术，但仅仅只了解技术是不够的。普通的技术人员和专家都了解技术，但是他们之间有很大的区别。专家能站在更高的层次审视设计，并能根据实际情况合适地选用技术。

专家是什么样的人？

- **专家们都很灵活**

专家知道正确、便捷的方法。并不是说他们会讨巧，干什么都想走捷径，而是说他们知道在什么时候该选用什么样的技术，如何安排人手、调用资源。由于他们对知识吃得很透，所以能够灵活地运用。有时候看上去很难的事情，由他们经手就会变得举重若轻。这是需要对工作有深刻的理解才能达到的境界。

- **专家们都很细心**

有时候看上去他们好像是在闲庭信步，但其实他们心细如发。由他们主导的工作很少会出现严重的纰漏，很多事情都能面面俱到，而且顾全大局。上了年纪的老马都会知道路上哪里有坑，但是年轻的马却很容易大大咧咧。这需要不断地积累经验才能做到。

- **但同时专家们又很大胆**

尽管专家非常谨慎细心，但是他们却又很有勇气跳出常规的圈圈。这种“大胆”不是盲目冒进，而是在有充分准备的基础上勇于创新，不排斥新兴技术。真正的专家是不会顽固不化的，如果是有真正价值和值得借鉴的新技术，他们一定很乐于接收。

- **最后，专家们也都很谦逊**

“大家风范”指的不是高高在上或者桀骜不驯，就连圣人都“不耻下问”。人们会发现其实他们很愿意和专家接触。哪怕你对某件事一窍不通，他们仍然会尊敬你，因为学得越多就越知道自己的不足，还有更多的知识需要掌握。也正因此专家们愿意接受新事物，并且乐于学习，始终保持着积极、高昂的姿态。

这就是专家。不管你是年轻的开发人员还是年长的设计师，这个世界都由你或即将由你来掌握。你们是中坚力量，世界的发展将由你们来推动。掌握已有的知识，熟练地运用，同时以开放的心态面对这个世界，它就在你的手中。

